



Reducción y ajuste de mallas triangulares

Edwar Samir Posada Murillo

Tesis presentada para obtener el grado de:
Magister en Ciencias - Matemática Aplicada

Directora:
Marianela Lentini Gil

Universidad Nacional de Colombia - Sede Medellín
Facultad de Ciencias, Escuela de Matemáticas

2013

Índice general

1. Preliminares	6
1.1. Mallas triangulares	6
1.1.1. Definición	6
1.1.2. Elementos y representación	6
1.1.3. Clasificación de vértices	9
1.1.4. Distancia Hausdorff	10
1.2. Configuración triangular de Brianchon	12
2. Marco general para reducción de mallas triangulares	16
2.1. Introducción y aspectos algorítmicos importantes	16
2.2. Operadores topológicos	17
2.2.1. Remoción de vértice	17
2.2.2. Colapso de medio lado	19
2.3. Nociones de distancia	21
2.3.1. Medida sobreestimada	22
2.3.2. Estimación directa del error global	23
2.4. Criterios de calidad	24
2.4.1. Funcional de calidad	25
3. Algunos algoritmos de reducción de mallas triangulares	27
3.1. Algoritmo SZL92	27
3.2. Algoritmo Sch97	28
3.3. Algoritmo KCS98	29
3.4. Otros algoritmos	30
3.5. Ejemplos	33

4. Construcción de parches triangulares cuadráticos racionales que yacen sobre cuádricas	36
4.1. Preliminares	36
4.2. Representación de Bézier de parches triangulares	37
4.3. Un algoritmo intuitivo para la construcción de parches cuadráticos racionales que yacen sobre cuádricas	38
4.3.1. El número de Ceva: factor heurístico	42
A. Algoritmos tipo “greedy”	48
B. Geometría diferencial de superficies	51
C. Curvas y superficies de Bézier	55

Introducción

Los polígonos representan figuras elementales muy populares en aplicaciones de geometría computacional y computación gráfica. Por un lado, tienen una representación matemática simple, y por otro lado, los paquetes de despliegue gráfico (hardware-software) comerciales trabajan principalmente con ellos, siendo muy común en esta área la representación de superficies geométricas por mallas poligonales. Dentro de este tipo de mallas, son las mallas triangulares las más usadas en animación y realidad virtual, debido a que estas aplicaciones involucran modelos geométricos que son usualmente representados por mallas triangulares. Sin embargo, debido a que la frontera de los polígonos es lineal a trozos y a que éstos son planos, se necesitan miles o millones de primitivas poligonales para capturar los detalles de alta complejidad geométrica, lo cual no es computacionalmente práctico ya que el tiempo y la memoria requeridos para el despliegue son proporcionales al número de polígonos. Como consecuencia, para que la visualización, modificación o manipulación de algunas mallas poligonales se pueda hacer en un tiempo computacional razonable, hace falta eliminar información “redundante” de la misma para reducir el número de triángulos pero manteniendo la calidad de aproximación. Los tomógrafos, escáneres de resonancia magnética, cámaras de alto rango, procesadores digitales de imágenes (elevación de terreno, satelitales), entre otros, usualmente generan mallas poligonales grandes.

La representación de superficies de Bernstein-Bézier ha ganado mucha importancia en el campo del Diseño Geométrico Asistido por Computadora (CAGD), y con éste, los parches triangulares de grado bajo ya que son simples desde un punto de vista matemático y computacionalmente convenientes en aplicaciones, especialmente aquellos que yacen sobre cuádricas pues son altamente requeridos en aplicaciones de ingeniería mecánica y arquitectura debido al rol de la superficie cuádrica. Así, para tener una representación más confiable de una superficie, desde un punto de vista geométrico, se pueden cambiar los triángulos planos de la malla por un tipo de parches triangulares que yacen sobre cuádricas, donde la construcción de estos parches depende de configuraciones dadas por los triángulos de la malla y las normales en sus vértices.

En el Capítulo 1 se estudian conceptos básicos sobre las mallas triangulares y se definen algunos términos usados a lo largo del documento: vecindad, distancia Hausdorff, número de Ceva, teorema de Brianchon, plano promedio entre otros.

En el Capítulo 2 se considera el marco general para los algoritmos de reducción de mallas trian-

gulares presentado por Kobbelt, Campagna y Seidel [5]. Se introducen los operadores topológicos, las nociones de distancia y los criterios de calidad como ingredientes fundamentales para construir un algoritmo de reducción de mallas triangulares.

En el Capítulo 3 se estudian los algoritmos de reducción de mallas triangulares presentados en [5], [7] y [8], entre otros, acompañados de ejemplos realizados con programas desarrollados en el transcurso de la tesis. En el Capítulo 4 se estudian las condiciones requeridas para la construcción de un parche triangular cuadrático racional de Bézier que yace sobre una cuádrica, basados en el algoritmo presentado en [1].

Finalmente, se bosqueja un algoritmo de reducción de mallas triangulares que genera mallas cuyos triángulos cumplen, en un alto porcentaje, las condiciones de existencia de un parche cuadrático triangular racional que yace sobre una cuádrica.

En los anexos se incluyen aspectos relevantes sobre geometría diferencial de superficies, algoritmos tipo “greedy” y superficies de Bézier.

Capítulo 1

Preliminares

1.1. Mallas triangulares

1.1.1. Definición

Una malla triangular es un tipo de malla poligonal que comprende un conjunto de triángulos, en nuestro caso en tres dimensiones, que se conectan por sus lados o vértices comunes y aproximan una superficie. Aunque el campo de aplicación de la generación automática de mallas triangulares ha sido tradicionalmente la obtención de modelos digitales de elevaciones del terreno, hoy en día son usadas en renderización¹, animación, realidad virtual y en análisis de elementos finitos.

1.1.2. Elementos y representación

Sea Ω una malla triangular. Los triángulos t_i de la malla cumplen:

- i. $\Omega = \cup_i t_i$
- ii. Cada triángulo es cerrado y con interior no vacío.
- iii. Para diferentes triángulos t_i y t_j la intersección de sus interiores es vacía.
- iv. Para diferentes triángulos t_i y t_j su intersección es un lado común, un vértice común o es vacía.

La mayoría de mallas triangulares almacenadas en la red o construidas por generadores de malla como Gmsh, son mallas triángulo-vértice, FV (Face-Vertex) por sus siglas en inglés, es decir, mallas

¹Generación de una imagen a partir de un modelo con ayuda de programas computacionales

triangulares que se representan por la conectividad de sus triángulos y sus vértices.

Una configuración como la mostrada en la Figura 1.1 se denomina **vecindad**, siguiendo el uso común de la topología dentro de un contexto discreto. Así, la vecindad de un vértice v_N es la colección de **triángulos incidentes** y **vértices adyacentes** a este. Los triángulos incidentes a v_N son aquellos triángulos que lo usan como uno de sus vértices, mientras los vértices adyacentes a v_N son los vértices distintos a este que son vértices de los triángulos incidentes. Por ejemplo, en la Figura 1.1, v_1, v_3, \dots, v_7 son los vértices adyacentes a v mientras t_1, t_2, \dots, t_7 son los triángulos incidentes.

La representación FV de una malla triangular usa dos matrices. Una de ellas, denominada **lista de vértices**, guarda las coordenadas de los vértices, y la otra, llamada **lista de triángulos**, almacena la definición de los triángulos de acuerdo a sus vértices. En la Figura 1.1 se muestra la representación FV de la vecindad de v .

Los algoritmos a estudiar requieren que cada vértice tenga asociada su vecindad, ya que sobre ésta actúa finalmente el algoritmo. Para eso, la lista de vértices se extiende con el almacenamiento de las vecindades, como se muestra en la Figura 1.2, donde para el primer vértice se almacena su vecindad orientada (orientación propia de cada vértice). Por facilidades computacionales las vecindades se representaran por un ciclo. Por ejemplo, el ciclo correspondiente a v es $v_1 v_2 \dots v_7 v_1$.

Una representación VV (vértice-vértice) de una malla triangular sólo usa la lista extendida de vértices.

Algunos algoritmos exigen que la representación de la malla triangular esté libre de información inútil: triángulos y vértices repetidos, triángulos nulos o degenerados que resultan de errores de transmisión y almacenamiento, producidos por los aparatos usados para tales fines. Como triángulos repetidos nos referimos no sólo a casos como $(v_l, v_p, v_n) = t_m = t_n = (v_l, v_p, v_n)$ sino también de la forma $(v_l, v_p, v_n) = t_m = t_n = (v_p, v_l, v_n)$.

Plano Promedio

Uno de los elementos asociados a los vértices de una malla triangular es el plano promedio. Dado que tres puntos en el espacio definen un plano, entonces cada triángulo de la malla tiene un plano asociado. Así, para el vértice v hay un plano por cada triángulo incidente. El plano promedio es entonces el “promedio” de estos planos asociados. Este plano se usa en todos los procesos de re-triangulación, como se verá en el capítulo siguiente. El problema de hallar el plano promedio asociado a un vértice v se entiende como el problema de hallar el plano que minimiza la distancia del conjunto de vértices de la vecindad al conjunto de planos en el espacio. Para encontrar tal plano se usan comúnmente dos técnicas: ajuste por mínimos cuadrados y análisis de componentes principales (ACP). Esta última es una técnica usada en estadística para reducir la dimensionalidad

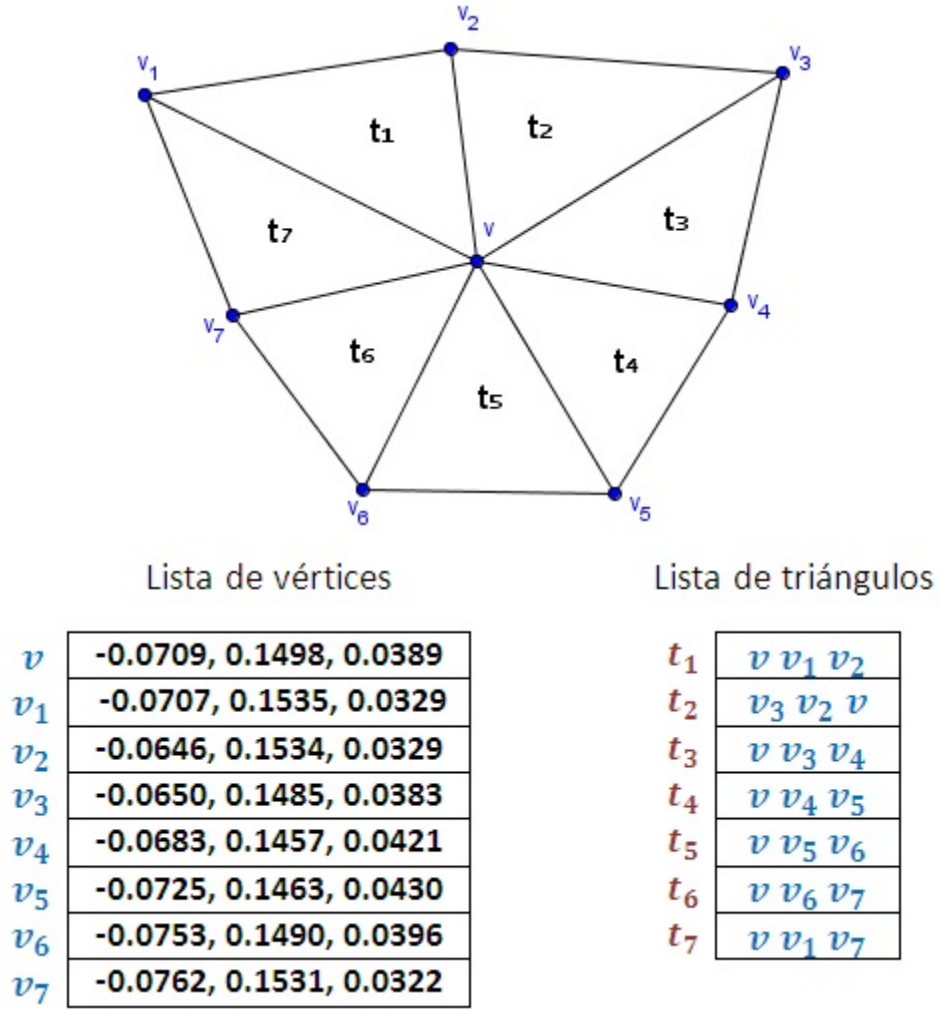


Figura 1.1: Vecindad de v , lista de vértices adyacentes y lista de triángulos incidentes.

v	-0.0709, 0.1498, 0.0389	$v_1 v_3 v_4 v_5 v_6 v_7 v_1$
v_1	-0.0707, 0.1535, 0.0329	$v_2 v v_8 \dots v_2$
v_2	-0.0646, 0.1534, 0.0329	$v_1 v v_3 \dots v_1$
v_3	-0.0650, 0.1485, 0.0383	$v_2 v v_4 \dots v_2$
v_4	-0.0683, 0.1457, 0.0421	$v_3 v v_5 \dots v_3$
v_5	-0.0725, 0.1463, 0.0430	$v_4 v v_6 \dots v_4$
v_6	-0.0753, 0.1490, 0.0396	$v_5 v v_7 \dots v_5$
v_7	-0.0762, 0.1531, 0.0322	$v_6 v v_1 \dots v_6$

Figura 1.2: vecindad de v en representación VV.

de un conjunto de datos y se describe de la siguiente manera:

Sea v un vector de la malla con n vértices adyacentes v_i .

- i. Se construye $R = \begin{pmatrix} v_1 - \bar{v} \\ \vdots \\ v_n - \bar{v} \end{pmatrix}$, matriz de covarianza, donde $\bar{v} = \frac{1}{n}\sum v_i$. Dado que \bar{v} es de dimensión 3, pues es el promedio de los vectores adyacentes de v , la dimensión de la matriz R es $n \times 3$
- ii. Se calcula el vector propio correspondiente al menor valor propio de la matriz $R^t R$.

Dicho vector propio es un vector normal del plano promedio.

Se escoge el método ACP ya que está implementado en MATLAB y arroja resultados confiables.

1.1.3. Clasificación de vértices

Los algoritmos de reducción de mallas triangulares que hacen parte de este estudio actúan sobre un vértice y su vecindad en cada paso de reducción, por eso es necesario clasificar los vértices de acuerdo a características topológicas y geométricas locales. La siguiente clasificación es la usada por Schroeder, Zarge y Seidel en [7].

Clasificación topológica

Sea S subconjunto de un espacio topológico, x pertenece al interior de S o x es un punto interior de S si existe una vecindad de x que está contenida en S . x está en la frontera de S o es un punto de frontera si y solo si toda vecindad suya contiene puntos de S y del exterior de S .

En el contexto discreto, un vértice **interior** es un vértice cuya vecindad está compuesta por un ciclo completo de triángulos, donde cada lado de la vecindad incide dos veces en el vértice, es decir, cada lado es usado por dos triángulos del ciclo. Un vértice **frontera** sólo posee un semiciclo de triángulos incidentes, es decir, hay dos lados en la vecindad que sólo inciden una vez en el vértice. Un vértice que no encaja en las clasificaciones anteriores se denomina **complejo** o no-variedad (non-manifold). Estos últimos deben ser tratados de forma cuidadosa para que los algoritmos de reducción de mallas triangulares sean robustos.

Con lo anterior también podemos clasificar, para fines prácticos, las vecindades como vecindad interior, vecindad frontera y vecindad compleja, dependiendo de la clasificación del vértice asociado. En la Figura 1.3 se observa la lista de triángulos incidentes al vértice v_{17} de la triangulación TBunny4 [12] y en la Figura 1.4 (c) su vecindad. Dicho vértice es complejo pues el lado $v_{17}v_{10}$ es usado por más de dos triángulos: t_{239} , t_{591} y t_{312} .

t_{239}	$v_{17} v_{101} v_{220}$
t_{591}	$v_{17} v_{101} v_{80}$
t_{94}	$v_{16} v_{17} v_{220}$
t_{160}	$v_{49} v_{17} v_{11}$
t_{213}	$v_{11} v_{17} v_{16}$
t_{312}	$v_{49} v_{101} v_{17}$

Figura 1.3: Triángulos asociados al vértice v_{17} .

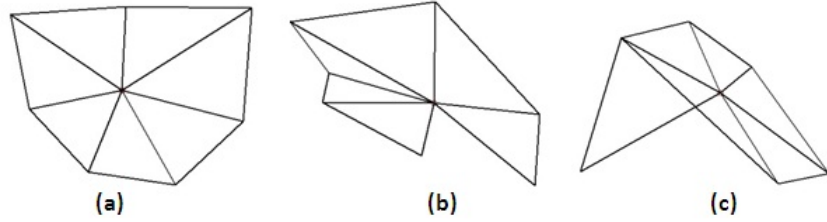


Figura 1.4: (a) Vecindad interior. (b) Vecindad frontera. (c) Vecindad compleja.

Clasificación geométrica

Un vértice, casi siempre interior, puede clasificarse geoméricamente de acuerdo a su curvatura local [Anexo B]. En el contexto de las mallas triangulares, la definición de curvatura se relaciona directamente con los ángulos diedrales de los triángulos de una vecindad. Dado que cada triángulo está sobre un plano, el ángulo diedral entre dos triángulos contiguos es el ángulo entre los planos que los contienen. Si el ángulo entre dos triángulos es mayor a un *ángulo característico* preestablecido, el lado común a esos dos triángulos se denomina *lado característico*. Los lados característicos señalan cambios bruscos de las normales presentes en la vecindad, es decir, cambios bruscos de planaridad. Así pues, una vecindad plana (o casi plana) no tiene lados característicos.

Cuando en la vecindad de un vértice se encuentran dos lados característicos, el vértice se denomina vértice de **lado interno**, y se denomina vértice **esquina** cuando se encuentra uno, tres o más lados característicos. De igual manera quedan clasificadas sus vecindades.

1.1.4. Distancia Hausdorff

El objetivo principal de las técnicas de reducción es obtener una malla triangular simplificada a partir de una malla inicial. Por lo tanto es importante contar con un método para medir la desviación de la nueva malla. La malla simplificada debe estar bastante “cerca” a la malla inicial

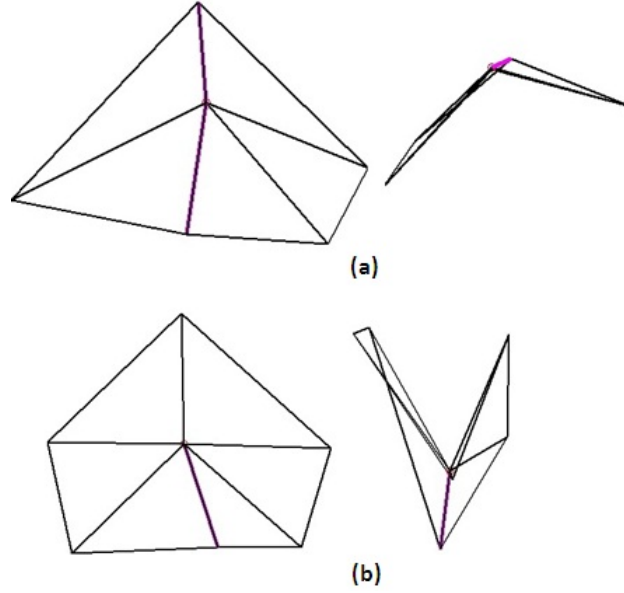


Figura 1.5: (a) Vecindad de lado interno y (b) Vecindad esquina desde diferentes perspectivas.

para seguir siendo una buena aproximación del objeto geométrico representado. La distancia euclídea entre un punto x y un conjunto $Y \subset \mathbb{R}^n$ está definida por

$$d(x, Y) = \inf_{y \in Y} d(x, y)$$

donde $d(., .)$ es la distancia entre dos puntos de \mathbb{R}^n . Usando esta definición, Klein, Liebich y Straßer definen en [4] la “distancia”: $d_E(X, Y)$ del conjunto X al conjunto Y como

$$d_E(X, Y) = \sup_{x \in X} d(x, Y) \quad (1.1)$$

la cual llaman **distancia Hausdorff de un lado** entre X e Y . Esta distancia Hausdorff de un lado no define una función distancia sobre todos los conjuntos de \mathbb{R}^n pues no es simétrica. Es decir, en general $d_E(X, Y) \neq d_E(Y, X)$. Si por ejemplo, la distancia Hausdorff de un lado $d_E(S, L)$ de la malla simplificada L a la malla inicial S es menor que una tolerancia ε , entonces

$$\forall x \in S \exists y \in L, d(x, y) < \varepsilon$$

Este resultado no es suficiente para controlar el proceso de reducción de mallas triangulares pues la no simetría de la distancia puede llevar a problemas como el mostrado en la Figura 1.6. Ya que si se quiere calcular la distancia Hausdorff de un lado de la malla simplificada a la inicial se podría violar la tolerancia establecida pues algunos puntos de la malla simplificada no están cerca de la malla inicial. Para solucionar ese tipo de problemas se introduce la **distancia Hausdorff**

$$d_H(X, Y) = \max(d_E(X, Y), d_E(Y, X)) \quad (1.2)$$

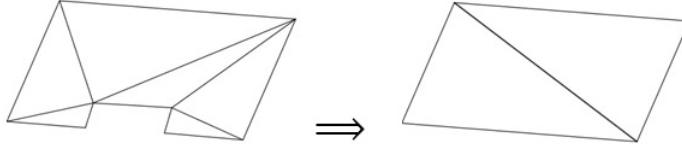


Figura 1.6: Mallas cuya distancia no puede ser calculada usando distancia Hausdorff de un lado.

La cual es una función distancia, cumpliendo $d_H(X, Y) = 0 \Leftrightarrow X = Y$. Así, si la distancia entre las mallas triangulares S y L es menor que una tolerancia ε entonces

$$\begin{aligned} \forall x \in S \exists y \in L, d(x, y) < \varepsilon \quad y \\ \forall y \in L \exists x \in S, d(x, y) < \varepsilon \end{aligned}$$

Klein et al resaltan también en [4] que para cualquier superficie parametrizada $S : \Omega \subset \mathbb{R}^2 \longrightarrow \mathbb{R}^3$ aproximada por una superficie lineal (plana) por tramos $L : \Omega \subset \mathbb{R}^2 \longrightarrow \mathbb{R}^3$ se tendrá

$$d_H(S, L) \leq \|S - L\|_\infty = \sup_{u \in \Omega} \|S(u) - L(u)\|$$

De donde, la distancia Hausdorff produce tasas de reducción mayores a las producidas por otras distancias para una misma tolerancia.

1.2. Configuración triangular de Brianchon

En algunas aplicaciones de computación gráfica y geometría computacional se requieren mallas triangulares con un gran número de triángulos para poder representar algunos aspectos geométricos de una superficie, lo cual no siempre es computacionalmente práctico. Por esta razón, es necesario reducir el número de triángulos de la malla para que sea menos compleja su manipulación, modificación o despliegue.

En la Figura 1.7 se observa una malla inicial de 1000 triángulos representando un conejo [12] y una malla reducida que consta de 414 triángulos. La malla simplificada es computacionalmente más manejable y gráficamente está bastante cerca a la superficie original, pero al seguir reduciendo la malla se pueden perder propiedades geométricas.

Así, para no perder la forma curva de ciertas zonas de la superficie, sería más conveniente cubrir la malla simplificada con triángulos curvos o parches triangulares en vez de triángulos planos. Un parche triangular, en este caso, cuadrático racional que yace sobre una cuádrica, se construye a partir de la configuración formada por un triángulo de la malla y las normales en sus vértices.

Para cada triángulo t_N de la malla reducida, sea π el plano generado por los vértices de t_N y τ el triángulo formado por las líneas de intersección de los planos definidos por las normales en los

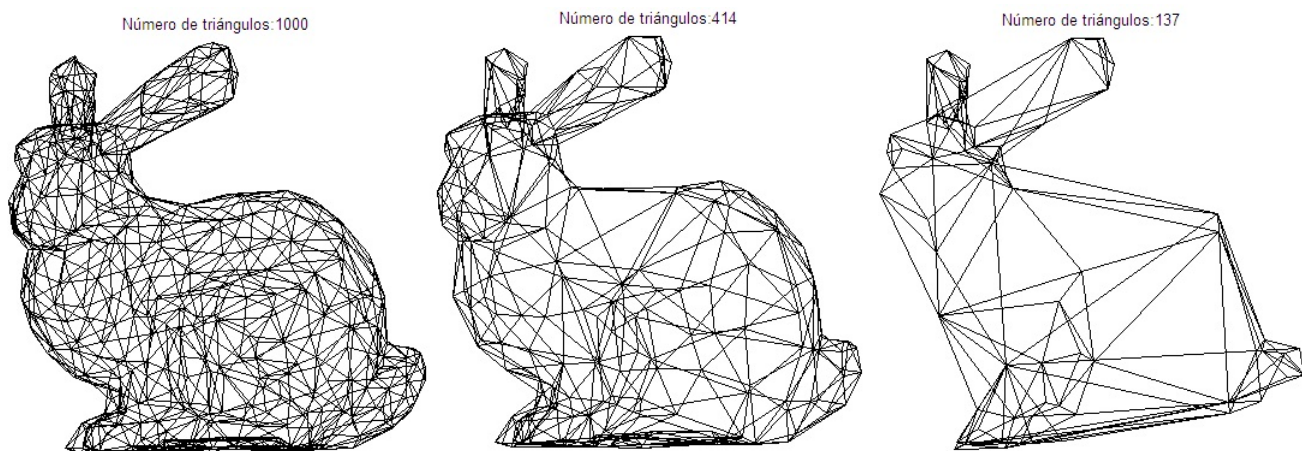


Figura 1.7: Triangulación Bunny5 y dos mallas reducidas.

vértices de t_N y el plano π . Luego, τ se encuentra sobre el plano π y junto con el triángulo t_N forman la configuración triangular mostrada en la Figura 1.8.

Para poder construir un parche triangular cuadrático racional que yazca sobre una cuádrica y tenga como planos tangentes en sus vértices los definidos por las normales en estos puntos, usando el algoritmo presentado en [1], la configuración formada por t_N y τ en el plano π debe ser una configuración de Brianchon.

Teorema 1.1 (Teorema de Brianchon)

Sea $ABCDEF$ un hexágono formado por seis rectas tangentes a una sección cónica. Los segmentos AD, BE, CF se intersecan en un solo punto

El punto de intersección se denomina punto de Brianchon. Si las tangentes AB, CD y EF coinciden con BC, DE y FA respectivamente, se obtiene una versión triangular del teorema de Brianchon. Es decir, en todo triángulo circunscrito a una sección cónica, las rectas que unen los vértices con los puntos de contacto de los lados opuestos concurren en un punto. En la Figura 1.9 se puede apreciar gráficamente las dos versiones del teorema.

Este resultado se puede generalizar como: dado un triángulo y una cónica tangente a sus lados o a las extensiones de sus lados, las rectas definidas entre los vértices y los puntos de tangencia opuestos son concurrentes.

Las rectas que, en un triángulo, pasan por un vértice y por un punto del lado opuesto se llaman cevianas, en honor al matemático italiano Giovanni Ceva. Las medianas, bisectrices y alturas de un triángulo son un tipo especial de cevianas.

Si se prescinde de la cónica, los resultados anteriores para triángulos quedan enmarcados en el

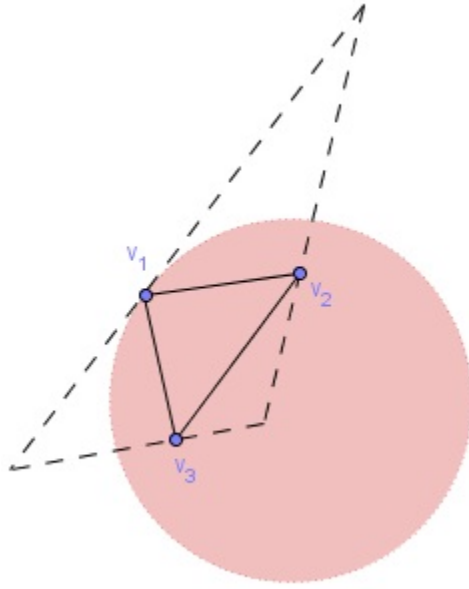


Figura 1.8: Configuración formada por τ , líneas punteadas, y t_N , líneas continuas, cuando los vértices de t_N están sobre una esfera.

siguiente teorema.

Teorema 1.2 (Teorema de Ceva)

Dado el triángulo ACE y los puntos B, D y F , sobre los lados AC, CE , y EA , o la extensión de estos respectivamente, las cevianas AD, BE y CF son concurrentes si y solo si

$$\frac{AB}{BC} \cdot \frac{CD}{DE} \cdot \frac{EF}{FA} = 1$$

El número de Ceva $\zeta = \frac{AB}{BC} \cdot \frac{CD}{DE} \cdot \frac{EF}{FA}$ es de gran importancia en los experimentos realizados con el algoritmo descrito en [1], los cuales se analizarán en el capítulo 4, para bosquejar un algoritmo de reducción y ajuste de mallas triangulares.

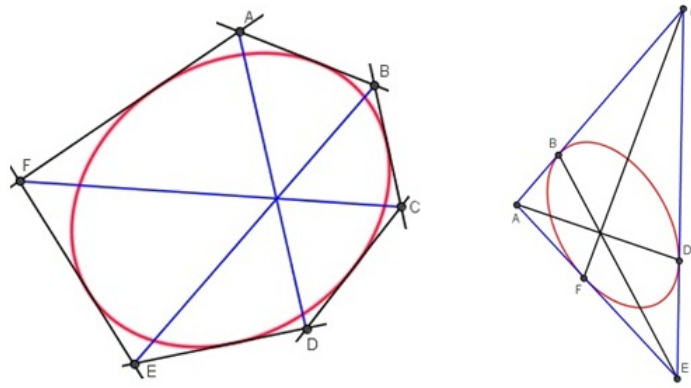


Figura 1.9: Versión hexagonal y triangular del teorema Brianchon.

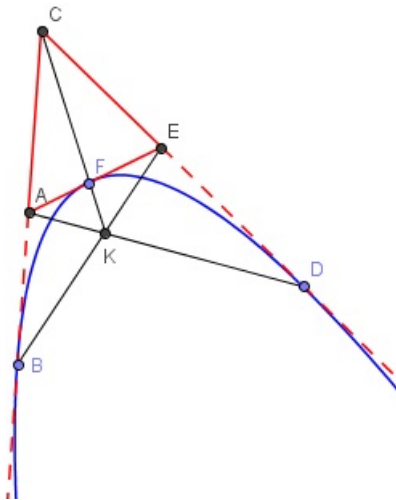


Figura 1.10: Configuración triangular de Brianchon cuando dos puntos de contacto están en la extensión de los lados del triángulo.

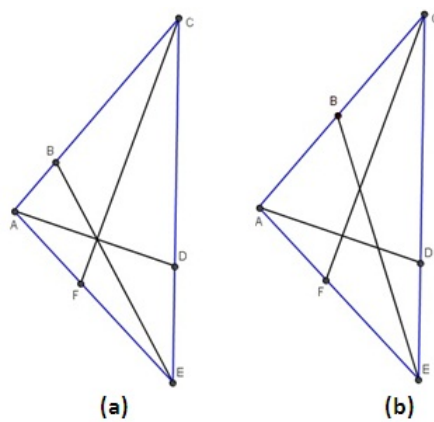


Figura 1.11: **(a)** Cevianas concurrentes. **(b)** Cevianas no concurrentes.

Capítulo 2

Marco general para reducción de mallas triangulares

2.1. Introducción y aspectos algorítmicos importantes

Para poder reducir el número de triángulos de una malla triangular se necesita eliminar información “redundante”. El término redundancia depende de la aplicación para la que se necesite trabajar con una malla triangular. En general, un vértice es redundante si se puede cambiar su vecindad por otra con un número menor de triángulos sin que esto altere notablemente las propiedades de la malla.

En aplicaciones gráficas, el término redundancia se asocia mucho con planaridad. Así, un vértice es redundante si su vecindad es casi plana. Por ejemplo, una malla triangular que conste de 1000 triángulos para representar un cubo está llena de información redundante, ya que tal cubo se puede aproximar perfectamente con tan solo 12 triángulos.

El algoritmo más famoso sobre reducción de mallas triangulares fue propuesto en 1992 por Schroeder, Zarge y Lorensen en [7]. Años más tarde, Kobbelt, Campagna y Seidel en [5] hacen un estudio sobre los algoritmos de reducción de mallas poligonales propuestos entre 1992 y 1997, concluyendo que todos estos algoritmos pueden ser vistos dentro de un mismo marco general. Por ejemplo, cada algoritmo de reducción está basado en un valor escalar que indica el grado de redundancia de un vértice, lado o triángulo en particular.

El objetivo principal de los algoritmos de reducción es construir una nueva malla a partir de una malla triangular inicial, de tal manera que esta nueva malla tenga un menor número de triángulos pero mantenga las propiedades geométricas de la superficie representada. Hay dos formas de lograr este objetivo: construir la malla simplificada alterando la topología de la malla inicial u obtener

la malla nueva por la aplicación de operaciones locales sobre la malla inicial pero sin cambiar su topología.

Preservar la topología de una malla limita a veces la capacidad total de reducción ya que superficies con un gran número de agujeros u otras características topológicas no pueden ser simplificados efectivamente. Además, tener un modelo topológicamente simplificado de una malla es útil en aplicaciones donde la topología por sí sola no arroja información importante sobre el problema a tratar. Así, en la renderización de algún objeto, pequeños agujeros pueden ser removidos sin afectar la calidad, pero en una simulación de elementos finitos sobre el mismo objeto, estos agujeros pueden ser importantes para obtener resultados confiables.

Los algoritmos que preservan la topología generalmente usan un operador simple que elimina una vecindad y re-triangula el agujero resultante. Algunos de estos algoritmos usan optimización para hallar la mejor re-triangulación.

Para controlar el proceso de reducción se debe dar un orden prioritario sobre el conjunto de vértices candidatos a ser eliminados. Este orden puede estar basado enteramente en nociones de distancia entre la malla inicial y la reducida u otras características de acuerdo a la aplicación. Para Kobbelt et al [5], esta descripción general concuerda con la mayoría de los algoritmos de reducción de mallas triangulares, por eso ellos identifican los tres “ingredientes” principales que se deben tener en cuenta a la hora de construir un algoritmo de reducción: **operadores topológicos** para modificar la malla localmente, **nociones de distancia** para garantizar que la nueva malla no esté “muy alejada” de la malla inicial y **criterios de calidad** para evaluar qué tan buena es la nueva malla.

2.2. Operadores topológicos

Los operados topológicos se pueden ver como funciones que, en la mayoría de los casos, eliminan un vértice y su vecindad para luego re-triangular el hueco resultante usando menos triángulos, reduciendo así el tamaño de la malla en cada iteración. El uso de los operadores depende de la clasificación del vértice, de una lista prioritaria y de cualquier otro recurso que tenga el algoritmo para decidir sobre qué vértice (lado, vecindad) se debe aplicar el operador. La mayoría de los operadores topológicos garantizan la no degeneración de la malla y, para preservar la topología de la misma, no se aplican sobre vértices y vecindades complejas.

2.2.1. Remoción de vértice

El clásico operador usado por Schroeder, Zarge y Lorensen en [7] remueve un vértice v junto a su vecindad y triangula el hueco resultante usando división recursiva. En cada iteración exitosa

una submalla de n triángulos es reemplazada por una nueva que sólo usa $n - 2$ triángulos si el vértice es interno o $n - 1$ si es de frontera. Este operador topológico actúa de diferentes maneras dependiendo el vértice.

Vértice interno

El operador topológico elimina un vértice y su vecindad asociada creando un agujero en la malla. Este agujero, proyectado sobre el plano promedio, tiene forma poligonal y lo representaremos por el ciclo de vértices que también representa la vecindad. La división de un ciclo se hace a lo largo de una línea (**línea divisora**) definida por dos vértices no consecutivos de este mismo. Cada ciclo resultante es dividido de nuevo hasta llegar a ciclos de tres vértices, es decir, triángulos. Estos triángulos formarán la submalla que reemplazará la vecindad eliminada. Las líneas divisoras yacen sobre planos ortogonales al plano promedio, llamados **planos divisores**.

En las Figuras 2.1 y 2.2 se observan dos de las diferentes posibles divisiones recursivas que pueden ser aplicadas a una vecindad de la malla triangular, las cuales dependen de la escogencia de las líneas divisoras y del orden en que se escojan. En ciertos casos, el operador no divide a lo largo de ciertas líneas pues al hacerlo se producirían triángulos cuyos interiores se intersecan, violando la definición de malla triangular dada en 1.1.2.

Durante el proceso de reducción, precisamente en la aplicación del operador topológico, se debe evitar la degeneración de la malla. Es decir, evitar la aparición de triángulos cuyos interiores se intersecan, triángulos con área cero o triángulos cuyos vértices son colineales, en general, triángulos que no cumplen las propiedades enunciadas en 1.1.2. Para detectar las divisiones que pueden degenerar la malla, el operador hace una comparación de medio espacio. Ya que una línea divisora cualquiera sobre la vecindad divide el ciclo en dos nuevos ciclos, la comparación de medio espacio determina si estos nuevos ciclos yacen en semiespacios diferentes definidos por el plano divisor, si esto sucede entonces la división vía esa línea divisora es válida. Por ejemplo, para el ciclo $ABCDEF A$ mostrado en la Figura 2.3, la división a lo largo de la línea BE no es válida pues el nuevo ciclo $EFABE$ no está contenido a un solo lado del plano divisor. Aunque esa primera división falla, una división hecha a lo largo de la línea DB funcionaría, pero en la mayoría de los casos el problema vuelve a presentarse en las divisiones siguientes ocasionando que no se pueda dividir el ciclo inicial y por ende que no se tenga una triangulación válida para éste.

Cuando existe más de una división válida el operador debe seleccionar la mejor, es decir, en cada proceso de división escoger las mejores líneas divisoras, actuando como un operador tipo “greedy” [Anexo A]. Para esto, se define la tasa r como la mínima distancia de los vértices del ciclo al plano divisor candidato dividida por la longitud de la línea divisora. Siendo el mejor plano divisor, y

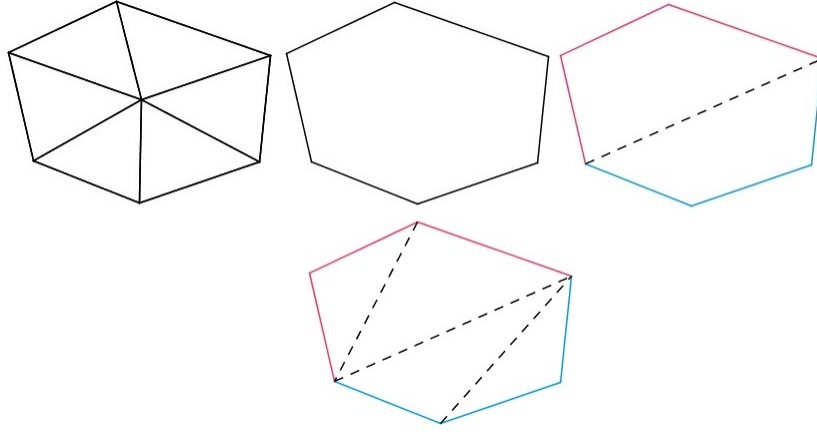


Figura 2.1: Proceso de división de una vecindad de la malla: se elimina el vértice y se triangula el ciclo resultante usando división recursiva.

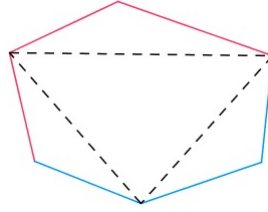


Figura 2.2: Otra posible triangulación para la vecindad inicial presentada en la Figura 2.1.

respectivamente mejor línea divisora, el que tenga la mayor tasa. Schroeder, Zarge y Lorensen [7] recomiendan una tasa mayor a 0.1 para tener un algoritmo robusto. De igual manera, se podría determinar otro valor, dependiendo de los requerimientos que la malla simplificada deba cumplir.

Vértice frontera y de lado interno

Si el vértice está en la frontera de la malla, el operador topológico completa el ciclo uniéndolo los dos vértices extremos y repite lo hecho para vértices internos. Para los vértices de lado interno el tratamiento es similar al de vértices internos. En este caso, el operador hace la primera división sobre la línea que definen los vértices de la vecindad que son extremo de los lados característicos, siempre y cuando la división sobre esta línea no produzca degeneración de la malla.

2.2.2. Colapso de medio lado

Kobbelt et al en [5] establecen que el operador topológico que usa un algoritmo de reducción de mallas triangulares no tiene un impacto significativo en la calidad final de dicha malla. En cambio, el criterio que usa el algoritmo para decidir donde aplicar dicho operador si es determinante para garantizar una malla de calidad. Ellos recomiendan trabajar con un operador topológico tan simple

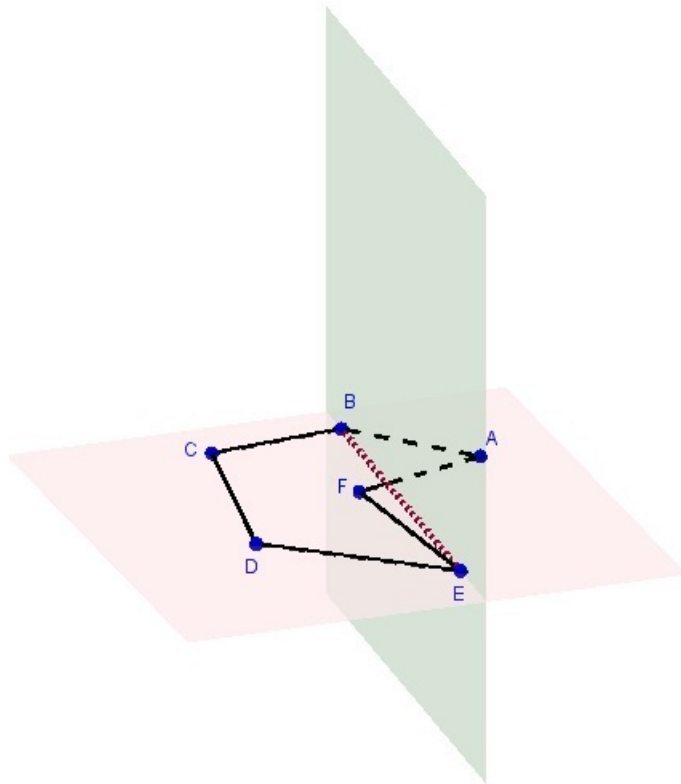


Figura 2.3: División no válida.

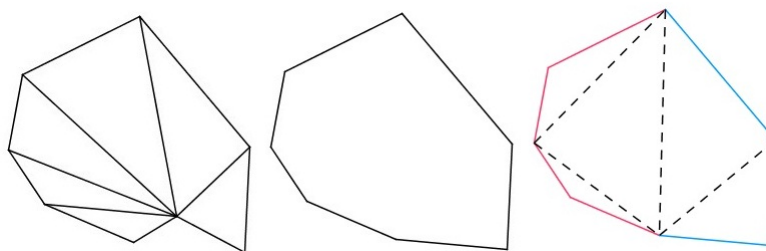


Figura 2.4: La vecindad inicial, por ser de frontera, es reemplazada por una vecindad con un triángulo menos y además elimina información redundante.

como sea posible, eliminando así todos los grados de libertad asociados a la geometría de la malla. Uno de los operadores topológicos más simples es el colapso de medio lado. Este operador colapsa el vértice candidato a eliminación sobre uno de los vértices de su ciclo, creando así una nueva triangulación para la vecindad. Este operador también reduce la malla en dos triángulos o en uno dependiendo de la clasificación de la vecindad.

El colapso de medio lado, a diferencia del operador descrito en 2.2.1, sólo debe escoger sobre qué vértice colapsar el vértice a eliminar, ya que el proceso de triangulación es inmediato. Aún así, hay que tener mucho cuidado en la escogencia del vértice colapsado para evitar que se presente degeneración en la malla. En la Figura 2.5 se observa un colapso aplicado sin alterar las propiedades básicas de la malla.

El vértice a eliminar v_S se colapsa sobre el vértice v_T si $v_S v_T$ es el menor lado que forma un colapso válido. Si hay lados característicos presentes es recomendable aplicar el colapso sobre el vértice extremo del menor lado característico, esto con el fin de acelerar el proceso de reducción. Para determinar si un colapso de medio lado es válido se debe hacer una comparación de medio espacio similar a la de 2.2.1. v_T junto con v_S forman un colapso válido si para todo vértice v_I no adyacente a v_T se garantiza que el plano que los contiene y es normal al plano promedio separa el ciclo en dos ciclos que no se solapan. Así, si el ciclo tiene n vértices, hay que hacer a lo más $n - 3$ comparaciones de medio espacio por cada vértice del ciclo para escoger los vértices que admiten un colapso válido. Por último el operador selecciona dentro de estos vértices el que esté a menor distancia de v_S y realiza el colapso.

Como en 2.2.1, si el vértice es de frontera se debe completar el ciclo y aplicar el operador como a las vecindades interiores.

2.3. Nociones de distancia

En la mayoría de aplicaciones técnicas que requieren la reducción de una malla triangular, existe una tolerancia predefinida que acota la desviación máxima de la malla reducida a la malla inicial. Para hacer un algoritmo útil de reducción de mallas triangulares hay que garantizar tal tolerancia sobre toda la superficie, descartando así los algoritmos que basan la decisión de si un vértice (lado, vecindad) es redundante o no teniendo en cuenta solamente características locales. Tal y como se puede medir distancia entre dos puntos o entre dos rectas en el plano, se puede medir distancia entre dos mallas triangulares. Es decir, existen métodos matemáticos que calculan qué tan “lejos” está una malla de otra. En este caso, se requiere medir la distancia entre la malla inicial y la malla resultante después de aplicar el proceso de reducción.

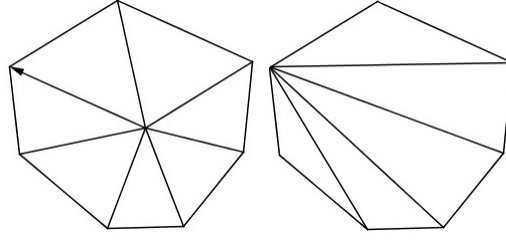


Figura 2.5: Colapso de medio lado.

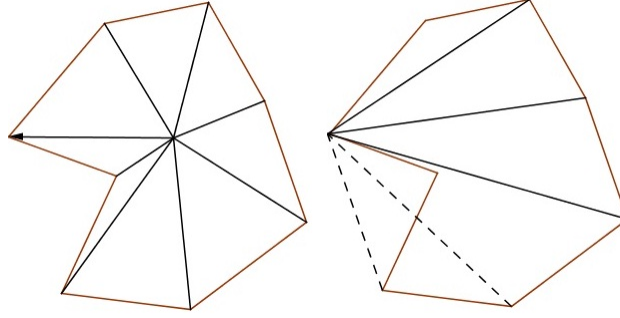


Figura 2.6: Colapso no válido.

La notación que en adelante usaremos es la dada por Ciampalini, Cignoni, Montani y Scopigno en [2]. Siendo S la malla inicial, S_i la malla intermedia obtenida después de aplicar el operador topológico i veces, L_v la vecindad del vértice v en la malla S_i y L'_v la vecindad que reemplazará a L_v en S_{i+1} después de la eliminación de v .

Bajo esta notación, ellos definen el error local y el error global. El primero mide el error de aproximación introducido al reemplazar L_v por L'_v , mientras el segundo mide el error de aproximación introducido cuando L'_v reemplaza a una submalla correspondiente de S . A medida que se avanza en el proceso de reducción, las vecindades se hacen más grandes, por lo tanto la vecindad L_v representa una submalla de S más grande, por ejemplo, en S_{20} que en S_3 . El error global exacto en el paso i se define como la distancia Hausdorff entre las mallas S y S_i .

El cálculo de esta distancia es computacionalmente costoso, por tanto para calcular errores globales de manera eficiente se usan algunas técnicas que aproximan la distancia Hausdorff. Dos de estas técnicas, una que sobreestima el error global partiendo de mediciones locales y otra que lo estima bastante bien basándose en la distancia Hausdorff de un lado, se estudiarán a continuación.

2.3.1. Medida sobreestimada

Esta primera técnica está basada en la acumulación de los errores locales introducidos después de cada aplicación del operador topológico.

Una de las técnicas presentadas en [2] por Ciampalini et al se centra en los triángulos de L_v y L'_v . Cada triángulo $t \in S$ tiene inicialmente error global cero. Cuando se aplica el operador topológico y se elimina el vértice v , se calcula el error local ε_l asociado a la vecindad. Luego, se selecciona el triángulo $t_k \in L_v$ con mayor error global y se suma el error global de ese triángulo a ε_l . El resultado de dicha suma será el error global de cada triángulo perteneciente a la nueva vecindad L'_v . El error global de cada vértice perteneciente a la malla S_i se define como el mayor error global de sus triángulos incidentes, mientras que el error global de la malla S_i será el mayor error global de sus triángulos. Así,

$$\varepsilon_g(t'_k) = \max_{t_k \in L_v} \varepsilon_g(t_k) + \varepsilon_l$$

$$\varepsilon_g(v_j) = \max_{t_r \in L_{v_j}} \varepsilon_g(t_r)$$

$$\varepsilon_g(S_i) = \max_{t \in S_i} \varepsilon_g(t)$$

donde ε_g representa al error global, t'_k es un triángulo de L'_v y v_j es un vértice de S_i . Esta técnica lleva a un estimativo grueso del error global real.

2.3.2. Estimación directa del error global

Si se quiere calcular la distancia exacta entre la malla inicial S y la malla intermedia S_i se debe usar la distancia Hausdorff, la cual requiere el cálculo de las dos distancias Hausdorff de un lado. Dado el alto costo computacional que requiere esto, algunos autores usan un método basado en dispersar puntos en ambas mallas y luego calcular la distancia euclídea entre ellos. Otra forma, presentada por Ciampalini et al en [2], calcula las distancias entre puntos dispersos en S y los triángulos de S_i , donde para fines prácticos los puntos dispersos se toman como los vértices de S .

Sea $v \in S$ un vértice que después de i pasos no ha sido eliminado, la distancia de v a los triángulos de S_i es cero, pues $v \in S_i$. Así, sólo interesa calcular la distancia de los vértices eliminados a los triángulos de S_i . Para hacer esto de forma eficiente hay que rastrear los vértices eliminados y los triángulos de la malla S_i que son cercanos a cada uno de estos vértices para hacer una asignación vértice eliminado - triángulo. A partir de esta asignación se puede calcular un estimativo del error global escogiendo la máxima distancia entre los vértices eliminados y sus triángulos cercanos.

En la iteración i -ésima, al eliminar el vértice v y cambiar la vecindad L_v por L'_v , se calcula el error global de la nueva malla como sigue:

- se selecciona el triángulo $t_k \in L'_v$ más cercano a v . Este vértice se asigna a la lista de vértices eliminados cercanos a t_k ,

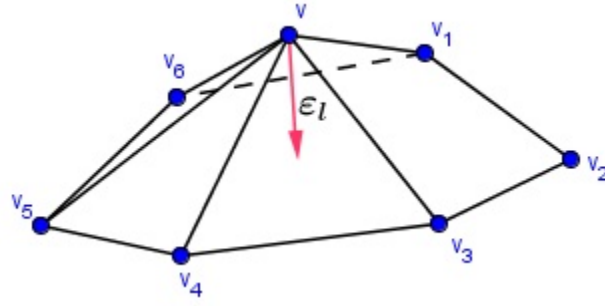


Figura 2.7: ε_l representa la distancia del vértice v al plano promedio. Dado que los vértices v_1, v_2, \dots, v_6 están tanto en L_v como en L'_v , el error local introducido al cambiar L_v por L'_v será la distancia de v a L'_v , la cual es aproximada por ε_l .

- se redistribuyen los vértices asignados a los triángulos de L_v entre los triángulos de L'_v ,
- se calculan las distancias entre los triángulos y sus vértices eliminados asignados.

El error global de un triángulo $t \in S_i$ se define como la máxima distancia entre éste y sus vértices eliminados asignados, mientras que el error global de S_i será el máximo error global de sus triángulos.

2.4. Criterios de calidad

Según Kobbelt et al en [5], si se formaliza el problema de reducir una malla triangular como un problema de datos dispersos, el objetivo central sería construir una superficie, en este caso una malla triangular S_i , cuya máxima distancia a los puntos dados P no viole una tolerancia predefinida.

$$\text{Dado un conjunto de puntos } P, \text{ construir una malla triangular } S \text{ tal que } \|S - P\|_{\infty} \leq \varepsilon. \quad (1)$$

Donde P es el conjunto de vértices de la malla inicial S .

Para Kobbelt et al [5], hallar una solución óptima global para el problema (1) es difícil, por lo tanto hay que conformarse con un mínimo local. Desde este punto de vista los algoritmos de reducción de mallas triangulares se asemejan a algoritmos tipo “greedy” [Anexo A] usados para resolver el problema de optimización descrito en (1). Se tiene por ejemplo, como en los problemas tipo “greedy”, una *función objetivo*, la cual cuenta el número de vértices eliminados, una noción de distancia que actúa como *función de factibilidad* para reflejar que tan lejos está una malla intermedia de exceder la máxima tolerancia y el *conjunto de candidatos* P . Por ejemplo, en muchos algoritmos de reducción se aplican primero aquellas reducciones que aportan el menor incremento en el error

de aproximación, lo cual es una estrategia tipo “greedy”, pues en cada iteración se está escogiendo la mejor opción local para obtener una solución óptima global.

Sin embargo, el problema de reducción de mallas triangulares es más complejo que un problema greedy, puesto que en el proceso de reducción las decisiones futuras dependen de decisiones anteriores, ya que la malla cambia cada vez que se aplica el operador topológico. Debido a esto, un criterio simple tipo “greedy” no obtendrá una buena aproximación a la solución óptima global. Se necesita entonces introducir un *criterio de calidad* que dirija el algoritmo, determinando sobre que vértices, y en qué orden, se debe aplicar el operador topológico.

Para Kobbelt et al [5] es importante usar conocimiento propio del problema, como el hecho que los puntos de P yazcan sobre una superficie razonablemente suave, por lo que S_i debe ser tan suave como sea posible. Es por esto que ordenar la lista de prioridad de acuerdo a la curvatura local [Anexo B] conlleva a buenos resultados. Esta lista se ordena de acuerdo a los efectos que tendría sobre la calidad de la nueva malla intermedia aplicar el operador sobre un determinado vértice.

2.4.1. Funcional de calidad

Aunque el funcional de calidad depende de la aplicación para la cual se necesite una malla simplificada, Kobbelt, Campagna y Seidel [5] desarrollan un funcional genérico que puede ser adaptado a objetivos particulares por el uso de factores de peso. Los operadores topológicos modifican la superficie localmente, por tanto, el funcional de calidad debe basarse en propiedades locales de la superficie. De la geometría diferencial de superficies [Anexo B] se sabe que la primera y la segunda forma fundamental caracterizan suficientemente bien, en muchas aplicaciones, el comportamiento de una superficie. En este contexto, la primera forma fundamental da cuenta de la distorsión local dentro de una superficie parametrizada, es decir, el mapeo de longitudes y ángulos, mientras la segunda forma fundamental proporciona información completa sobre las curvaturas locales. Estos conceptos pueden transferirse al campo discreto de las mallas triangulares, ya sea por interpolantes poligonales de grado bajo o analogías geométricas llevadas a caracterizaciones equivalentes.

Si se asignan valores paramétricos (r_i, s_i) a los vértices adyacentes $v_k \in L_v$, se puede construir un polinomio cuadrático que ajuste los parámetros localmente usando mínimos cuadrados. Los coeficientes de este polinomio pueden considerarse como los coeficientes de Taylor de una expansión local. Teniendo información hasta segundo orden de las derivadas es suficiente para construir las formas fundamentales localmente, lo cual da acceso a aproximaciones de propiedades geométricas importantes. Pero, este proceso se puede optimizar usando analogías geométricas. Por ejemplo, la redondez de los triángulos (la relación entre el radio del círculo inscrito y el mayor lado) puede ser usada para estimar la distorsión local y el ángulo diedral entre triángulos adyacentes para medir

la curvatura local.

Combinar estas propiedades locales de superficie en el funcional

$$F(S) := \sum_{v \in S} (\alpha E(v) + \beta R(v) + \gamma H(v)) \quad (2.1)$$

permite dar mayor prioridad a $E(v)$, a la distorsión local $R(v)$ o a la curvatura local $H(v)$ simplemente ajustando los coeficientes de peso α, β y γ . Donde $E(v)$ se define como el error local o global, $R(v)$ como la redondez promedio de los triángulos de L_v y $H(v)$ como la suma de los ángulos diedrales entre triángulos de la vecindad.

Por ejemplo, si la malla reducida debe preservarse con tanto detalle como sea posible, con un número preestablecido de triángulos, α debe ser un valor grande. Un valor grande de β conlleva a mallas reducidas que son apropiadas para el análisis de elementos finitos, ya que en éste se prefieren triángulos no muy agudos. Si γ es el coeficiente mayor entonces la malla será optimizada con respecto a calidad geométrica, la cual es importante en renderización.

En algunos casos es conveniente usar un funcional de calidad en cascada: para comparar dos reducciones potenciales se puede comparar primero su impacto sobre $H(v)$, si estos son aproximadamente iguales entonces se puede comparar el valor de $R(v)$ y así sucesivamente.

Capítulo 3

Algunos algoritmos de reducción de mallas triangulares

3.1. Algoritmo SZL92

Schroeder, Zarge y Lorensen fueron los primeros en presentar un algoritmo que no reordena toda la malla triangular para reducir el número de triángulos, sino que usa un operador simple que actúa localmente y reduce la malla en uno o en dos triángulos por cada iteración. Estos tipos de algoritmos de reducción son llamados incrementales. Con su algoritmo logran *“reducir el número total de triángulos en una malla triangular, preservando la topología inicial y manteniendo una buena aproximación a la geometría inicial”*

Operador

Este algoritmo usa el operador remoción de vértice descrito en 2.2.1. El algoritmo no aplica el operador sobre vértices complejos o esquina, aunque si la aplicación requiere un gran porcentaje de reducción, los vértices esquina y de lado interno pueden ser tratados como vértices internos. Esto debido también a que no siempre es deseable retener ángulos característicos. En ocasiones, las mallas contienen zonas de triángulos muy pequeños con grandes ángulos característicos que contribuyen poco a la aproximación geométrica, por ejemplo, triángulos pequeños que hacen parte del ruido generado por los aparatos y dispositivos que participan en la construcción de la malla.

Noción de distancia

Este algoritmo sólo tiene en cuenta el error local generado al eliminar un vértice. No mide el error global sobre los vértices ni mide la distancia entre la malla inicial y mallas intermedias.

El algoritmo fue pensado inicialmente para usarse en la visualización de imágenes generadas por

tomógrafos, médicos e industriales, y por modelos digitales de terrenos. El algoritmo es netamente intuitivo, si la malla reducida no es de total agrado, se debe correr el algoritmo modificando el valor de la tasa, operar sobre vértices esquinas o modificar el porcentaje de reducción.

Criterio de calidad

Este clásico algoritmo fue diseñado para tratar con mallas triangulares con un número muy grande de triángulos. El interés de los creadores era poder reducir de manera drástica el número de triángulos para poder manipular los datos contenidos en la malla, sin darle mucha importancia a las sutilezas geométricas de la superficie o al buen aspecto de los triángulos de la malla resultante. La lista de prioridad de este algoritmo se rige, primero por la clasificación del vértice, descartando a vértices complejos y esquina, y luego por el error local. En el contexto del Funcional de calidad tendríamos un peso positivo para $E(v)$ y pesos nulos para $H(v)$ y $R(v)$.

3.2. Algoritmo Sch97

El algoritmo presentado por Schroeder en 1997 no sólo reduce una malla triangular sino que, de ser necesario, modifica la topología de la malla, y almacena progresivamente las mallas intermedias.

Operador

Este algoritmo posee dos operadores topológicos. El operador principal es colapso de medio lado, detallado en 2.2.2, encargado de reducir la malla y preservar la topología inicial, mientras el operador divisor de vértice puede modificar la topología en cada iteración.

El operador divisor de vértice se encarga de dividir aquellas vecindades en la que no se puede aplicar un colapso de medio lado, ya sea porque son vecindades complejas o porque su forma no permite colapsos válidos. Esta división se obtiene al remplazar el vértice v_x , candidato a eliminación, por un vértice v_s en la lista de conectividad de algunos triángulos de su vecindad, creando así una separación ficticia. Finalmente se colapsan v_s y v_x sobre uno de los vértices que conforman su nueva vecindad. En la Figura 3.1 se observa la aplicación del operador división de vértice sobre una vecindad interior.

Noción de distancia

Este algoritmo usa una medida sobreestimada similar a la detallada en 2.3.1 para calcular la distancia entre la malla inicial y una malla intermedia. La diferencia está en que este algoritmo no acumula los errores sobre los triángulos sino sobre los vértices de la malla intermedia, calculando el error global de la malla como el máximo error global de los vértices. Así,

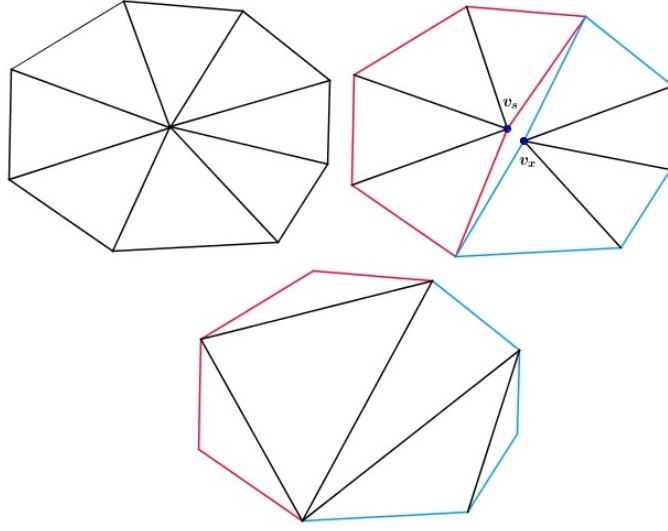


Figura 3.1: A una vecindad a la cual no se le pudo aplicar colapso de medio lado se le aplica el operador divisor de vértice.

$$\varepsilon_g(v_j) = \varepsilon_g(v_j) + \varepsilon_l$$

$$\varepsilon_g(S_i) = \max_{v_j \in S_i} \varepsilon_g(v_j)$$

donde v_j es un vértice de la malla intermedia S_i y ε_l es el error local introducido al eliminar un v y cambiar L_v por L'_v .

Criterio de calidad

En este algoritmo, como en el anterior, sólo se le da peso al factor $E(v)$ que representa, en este caso, el error global del vértice v . Además, los vértices solitarios deben ser eliminados primero. Es decir, los vértices frontera cuya vecindad sólo posee dos triángulo son ubicados a la cabeza de la lista. Esto se hace cambiando la forma de medir el error local de estos vértices, el cual será la raíz cuadrada del área del único triángulo de su vecindad, en vez de la técnica aplicada para el resto de vértices.

3.3. Algoritmo KCS98

Kobbelt, Campagna y Seidel [5] diseñaron un algoritmo de reducción de mallas triangulares que encaja perfectamente en el marco general creado por ellos, estudiado en el capítulo 2, ya que posee un operador topológico simple, una buena estimación de los errores globales y un criterio de calidad claro.

Operador

El operador usado por este algoritmo es el colapso de medio lado. Esta elección es importante no solo por la simplicidad de este operador topológico sino también porque es fácilmente invertible, permitiendo así el almacenamiento progresivo de los colapsos, y con esto, de las mallas intermedias sobre una vecindad interior.

Noción de distancia

Este algoritmo usa una estimación directa del error global.

Un cálculo aproximado de la distancia Hausdorff de un lado entre el conjunto original de vértices y la malla simplificada es computacionalmente costoso, por eso se tienen en cuenta ciertas características que permiten optimizarlo, como el hecho que la malla es modificada usando un colapso de medio lado.

Cuando el lado \overline{qp} es colapsado en el vértice p , se debe calcular la distancia entre q y la malla resultante. Para reducir la complejidad de esta operación se restringe el área de búsqueda, del punto de mínima distancia, a la submalla afectada por el colapso.

Como se mencionó en 2.3.2 cada vez que se hace una reducción, el vértice eliminado debe ser asignado a su triángulo más cercano de la nueva vecindad. Para que esta distribución sea eficiente, se deben hacer comparaciones de medio espacio utilizando los lados de los triángulos de la vecindad para determinar sobre qué triángulos se encuentran estos vértices. Finalmente, se calcula la distancia del vértice eliminado a su triángulo asignado. Donde el error global de cada triángulo de la malla S_i es la máxima distancia del triángulo a sus vértices eliminados asignados, y el error global de la malla es el máximo error global de sus triángulos.

Criterio de calidad

En una fase inicial todos los posibles colapsos deben ser evaluados y clasificados de acuerdo al funcional de calidad. Solamente los colapsos válidos deben ser ubicados en la lista de prioridad en orden de calidad decreciente. Después de aplicar un colapso la lista debe ser actualizada reevaluando los posibles colapsos de la zona afectada por la aplicación del operador.

3.4. Otros algoritmos

Los siguientes algoritmos de reducción de mallas triangulares, aunque también encajan en el marco general definido por Kobbelt et al en [5], son claramente diferentes a los citados anteriormente.

El algoritmo presentado por Garland y Heckbert en [9] usa como operador topológico la contracción de vértices y una noción de distancia basada en un error cuadrático. El algoritmo presentado por Hussain, Okada y Nijima en [10] es bastante reciente (2004) y según sus autores, tiene un mejor desempeño que los algoritmos más notables de ese tipo presentados hasta ese entonces. Este algoritmo usa como operador topológico el colapso de medio lado e implementa una noción de distancia basada en aspectos visuales.

Algoritmo GH97

El operador topológico usado por este algoritmo es una contracción de vértices, es decir, los vértices v_i y v_j se colapsan en un nuevo punto v . La diferencia de este operador con el colapso de lado, el cual colapsa dos vértices en un nuevo vértice, es que para usar contracción de vértices no es necesaria adyacencia previa entre v_i y v_j . Esto conlleva a que zonas separadas se junten, es decir, pasar posiblemente de una malla disconexa a una conexa. Este operador se aplica solamente si (v_i, v_j) es una pareja válida, es decir, si es un lado o si $\|v_i - v_j\| < \epsilon$ donde ϵ es un valor establecido.

Para decidir cual contracción realizar en una determinada iteración de algoritmo, se debe asociar a cada vértice una matriz Q_v simétrica de dimensión 4×4 . Se define el error en cada vértice $[v]$ como la forma cuadrática $\Delta(v) = v^T Q_v$. Así, para realizar la contracción $(v_i, v_j) \rightarrow \bar{v}$, se debe escoger la posición de \bar{v} , y no solo esto, si no que se debe escoger de tal manera que $\Delta(\bar{v})$ sea mínima. Dado que $\Delta(v)$ es cuadrática, hallar su mínimo es un problema lineal, el cual es equivalente a resolver

$$\begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{12} & q_{22} & q_{23} & q_{24} \\ q_{13} & q_{23} & q_{33} & q_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \bar{v} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

donde q_{ij} son las coordenadas de la matrix $Q_{\bar{v}} = Q_i + Q_j$.

Si el sistema anterior no tiene solución entonces se intenta hallar \bar{v} en el segmento $v_i v_j$.

El algoritmo tiene la siguiente descripción:

- Calcule la matriz Q_v para todos los vectores iniciales.
- Seleccione todos los pares (v_i, v_j) válidos.
- Calcule el vector óptimo \bar{v} sobre el cual realizar la contracción para cada par válido. $\Delta(\bar{v})$ es el costo de realizar la contracción.
- Organice la lista de prioridad de acuerdo al valor de $\Delta(\bar{v})$, de menor a mayor.

- Realice las contracciones de acuerdo a la lista de prioridad y actualice la información en cada iteración.

Las matrices iniciales Q_v se calculan usando los planos definidos por los triángulos de la vecindad de v .

Para un vértice v de una malla triangular, hay tantos planos p como triángulos en su vecindad. Este plano está representado por los coeficientes de su ecuación normalizada $p = [a, b, c, d]$ con $a^2 + b^2 + c^2 = 1$. Así, se define la matriz $K_p = pp^T$ y la matriz $Q_v = \sum K_p$.

Algoritmo HON04

Hussain et al [10] usan como operador topológico el colapso de medio lado y dan mucha importancia a lo que ellos llaman la importancia visual del vértice.

La importancia visual de un vértice v de la malla se define como $w_v = 1 - \|k_v\|$ donde $k_v = \frac{\sum \Delta_i \eta_i}{\sum \Delta_i}$, η_i es la normal y Δ_i el área de un triángulo t_i de la vecindad de v . La suma se hace sobre los triángulos de esta vecindad.

Cuando se realiza el medio colapso del vértice v en unos de sus vértices adyacentes v_j los triángulos que no tienen como vértice a v y a v_i simultáneamente sufren una transformación, pasan de ser $t_i = (v, v_k, v_l)$ a $t'_i = (v_j, v_k, v_l)$. De acuerdo a esto, se define para estos triángulos $Q_{t_i} = l_{t_i} \theta_{t_i}$, donde $l_{t_i} = 0.5(\Delta_i - \Delta_{i'})$ y θ_{t_i} es el ángulo descrito por la normal unitaria del triángulo t_i al rotar sobre el lado $v_k v_l$. El costo de colapsar a v sobre v_j se define como la suma, sobre los triángulos descritos arriba, de los Q_{t_i}

El algoritmo está descrito como sigue:

- Calcule la importancia visual w_v de cada vértice v de la malla.
- Para cada vértice de la malla compute los costos de colapsar a v sobre sus vértices adyacentes. Seleccione el colapso que tenga menor costo. De acuerdo a la importancia visual, ubique el vértice en la lista de prioridad.
- Ejecute el primer colapso de la lista de prioridad y actualice los valores de importancia visual de los vértices afectados por el colapso.
- Repita los dos últimos pasos hasta que no haya más colapsos en la lista de prioridad.

3.5. Ejemplos

Los algoritmos mencionados en las secciones 3.1., 3.2., y 3.3. fueron usados para crear un programa (llamado Decimat) en MATLAB que reduce una malla triangular cualquiera permitiendo la variación de los operadores topológicos y las nociones de calidad por parte del usuario. Se programaron los operadores topológicos remoción de vértice y colapso de medio lado, se creó el programa que transforma una malla triangular FV en una malla VV y el programa que actualiza la vecindad a medida que se aplica el operador topológico, entre otros programas auxiliares.

En la Figura 3.2 se muestra, en la parte superior izquierda, la malla triangular inicial TBunny4 [12] que consta de 948 triángulos. La malla triangular a su derecha es una malla generada por Decimat usando el operador topológico remoción de vértice y teniendo en cuenta solamente el criterio de calidad $E(v)$ (aproximación local). La malla triangular ubicada en la parte inferior izquierda de la imagen fue creada bajo la guía del criterio de calidad $R(v)$ (redondez promedio) mientras para la creación de la malla triangular inferior derecha sólo se tuvo en cuenta el criterio de calidad $H(v)$ (curvatura local).

Las diferencias entre las tres mallas generadas por Decimat son sutiles. Ese es precisamente el objetivo del funcional de calidad, que a partir de una malla inicial se generen mallas triangulares reducidas que puedan usarse en aplicaciones diferentes como renderización o análisis de elementos finitos, sin que se pierda la forma original del objeto modelado.

Para la Figura 3.3 establecimos una reducción a 140 triángulos a partir de una malla inicial de 304 triángulos usando diferentes criterios de calidad. En la construcción de la malla reducida del centro sólo se tuvo en cuenta el criterio de calidad $E(v)$, mientras en la malla derecha se buscó que los triángulos resultantes tuviesen una buena redondez ($R(v)$). Finalmente, los triángulos de la malla triangular izquierda son redondos y serían útiles en un análisis de elementos finitos, más que los triángulos de la malla central, pero en la búsqueda de una malla con un buen promedio de redondez en sus triángulos, se llegó a una malla que no aproxima bien a la malla inicial o al objeto modelado, algo que no sucede con la malla central. Este problema se soluciona requiriendo un menor número de triángulos para la malla izquierda o combinando los criterios de calidad, en un solo funcional o en cascada, como se planteó en el capítulo 2. A pesar de lo observado en la Figura 3.3, los operadores topológicos remoción de vértice y colapso de medio lado preservan las restricciones topológicas de la malla inicial y por ende del objeto modelado, siempre y cuando se trabaje con valores adecuados. Por ejemplo, la malla ubicada en la parte superior derecha de la Figura 3.4 es una malla reducida a partir de una malla inicial de 664 triángulos usando remoción de vértice, con proceso de reducción guiado por el criterio de calidad $E(V)$. Durante el proceso de reducción, el agujero del centro de la malla se hizo más pequeño aunque no se cerró, pero si en

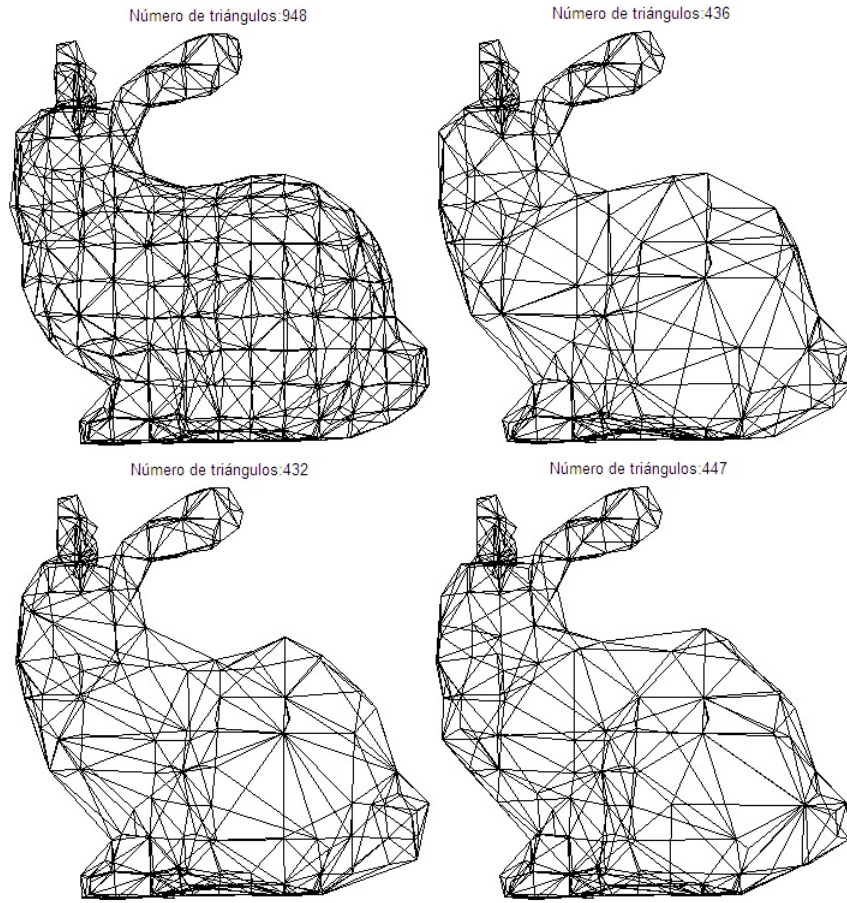


Figura 3.2: (**Arriba**) Malla triangular inicial. Malla reducida bajo criterio $E(v)$. (**Abajo**) Malla reducida bajo criterio $R(v)$. Malla reducida bajo criterio $H(v)$.

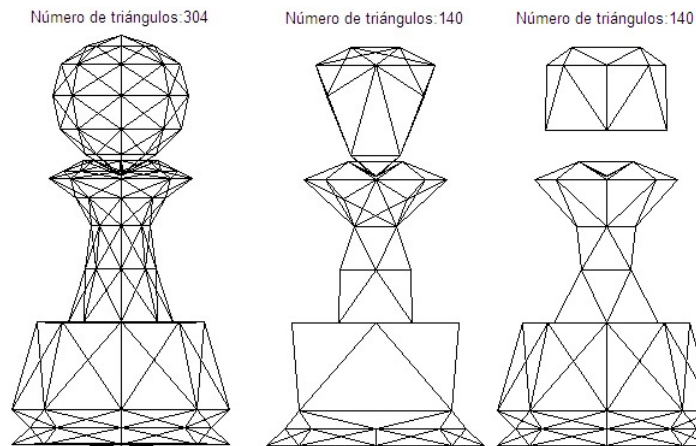


Figura 3.3: Malla original y mallas reducidas usando criterios $E(v)$ y $R(v)$.

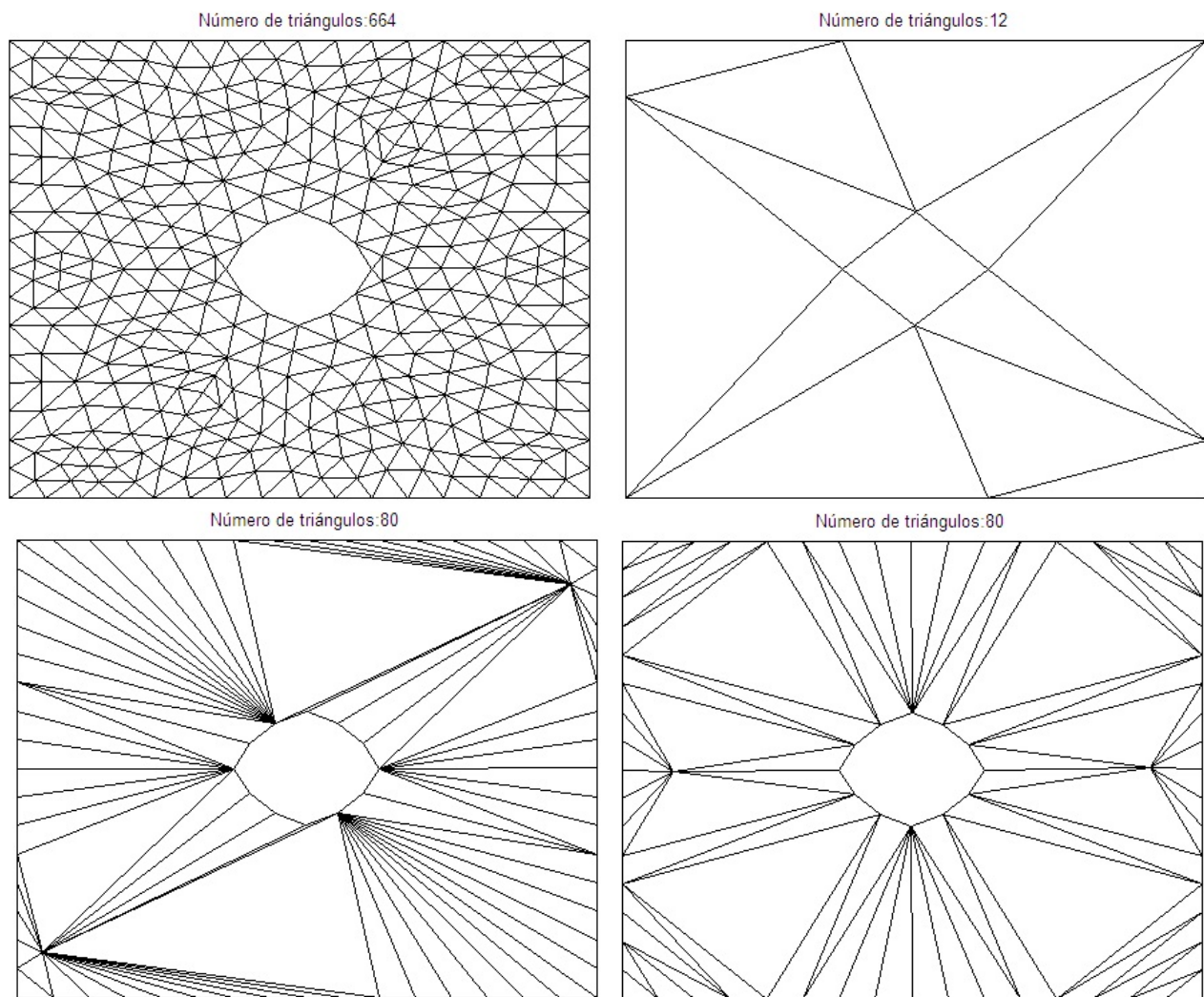


Figura 3.4: Malla original y mallas reducidas usando operadores topológicos remoción de vértices y colapso de medio lado.

una nueva corrida, se usa una cota grande para el error de aproximación $E(V)$ en los vértices que conforman la frontera interna, el hueco tenderá a cerrarse completamente.

En la parte inferior de la figura, se observan mallas triangulares de 80 triángulos, reducidas a partir de una malla inicial usando como operador topológico remoción de vértice y colapso de medio lado respectivamente.

Capítulo 4

Construcción de parches triangulares cuadráticos racionales que yacen sobre cuádricas

Paul de Faget de Casteljau inventó las curvas de Bézier en 1959 y en esa misma fecha se percató de la necesidad de extender sus ideas a superficies (ver Anexo C sobre curvas y superficies de Bézier). El primer tipo de superficies que consideró fue lo que hoy se conoce como triángulos de Bézier.

4.1. Preliminares

El algoritmo presentado por Albrecht, Paluszny y Lentini en [1] construye parches triangulares cuadráticos racionales que yacen sobre cuádricas a partir de la información presente en los triángulos de una malla triangular. Las entradas del algoritmo son las coordenadas 2D de tres vértices no colineales y de sus normales.

En la sección 1.2 se define, para cada triángulo t_N de una malla reducida, el plano π como el plano generado por los vértices de t_N y en ese plano se considera el triángulo τ , formado por las líneas de intersección de los planos tangentes a la superficie en los vértices y el plano π .

La concurrencia de las cevianas formadas por los vértices del triángulo τ y los vértices de t_N (ver Figura 1.9) es condición suficiente para que una configuración de vértices y sus respectivas normales sea compatible con la construcción de un parche triangular cuadrático racional que yace sobre una cuádrica, que interpola los vértices dados y tiene como planos tangentes en los vértices los planos definidos por las normales dadas.

Albrecht, Paluszny y Lentini [1] usaron las restricciones impuestas por el teorema de Brianchon para construir dichos parches. Cuando una configuración dada de vértices y normales a la superficie en éstos no es una configuración de Brianchon, se aplica un proceso de optimización que modifica las normales para que la nueva configuración vértices-normales sea de Brianchon y admita la construcción de un parche triangular. El parche triangular construido de esta manera minimiza la desviación de las condiciones de tangencia. Es decir, se determina en la familia de todos los parches triangulares cuadráticos racionales que yacen sobre una cuádrica e interpolan los vértices dados, aquel que minimice la desviación de las condiciones de tangencia.

4.2. Representación de Bézier de parches triangulares

Según Paluszny, Prautzsch y Boehm [6], la mejor representación de los polinomios de Bernstein (Anexo 3), debido a su simetría, se da en coordenadas baricéntricas.

Dado un triángulo con vértices a_0, a_1, a_2 en \mathbb{R}^2 , las coordenadas baricéntricas de un punto $x \in \mathbb{R}^2$ con respecto a ese triángulo se definen como $\chi = (x_0, x_1, x_2)$, donde $x = x_0a_0 + x_1a_1 + x_2a_2$ y $x_0 + x_1 + x_2 = 1$.

Usando esta última propiedad se llega a la siguiente igualdad,

$$1 = (x_0 + x_1 + x_2)^n = \sum_{i+j+k=n} \frac{n!}{i!j!k!} x_0^i x_1^j x_2^k, \quad 0 \leq i, j, k$$

El desarrollo anterior se usa para definir los **polinomios de Bernstein**:

$$B_{ijk}^n(x_0, x_1, x_2) = \frac{n!}{i!j!k!} x_0^i x_1^j x_2^k, \quad i + j + k = n$$

donde n determina el grado de los polinomios. Así, los polinomios de Bernstein de grado 2 son $B_{002}^2, B_{101}^2, B_{011}^2, B_{200}^2, B_{020}^2, B_{110}^2$. Estos polinomios son una partición de la unidad.

Un **triángulo de Bézier** o **parche triangular** definido por los puntos de control \mathbf{b}_{ijk} sobre un triángulo de referencia $a_0a_1a_2$ se define como

$$\mathbf{b}(x) = \sum_{i+j+k=n} \mathbf{b}_{ijk} B_{ijk}^n(x_0, x_1, x_2)$$

donde $\chi = (x_0, x_1, x_2)$ son las coordenadas baricéntricas de x .

Usando las notaciones $\chi = (x_0, x_1, x_2)$, $\mathbf{i} = (i, j, k)$ y $|\mathbf{i}| = i + j + k$, la definición anterior se puede escribir como

$$\mathbf{b}(x) = \sum_{|\mathbf{i}|} \mathbf{b}_{\mathbf{i}} B_{\mathbf{i}}^n(\chi)$$

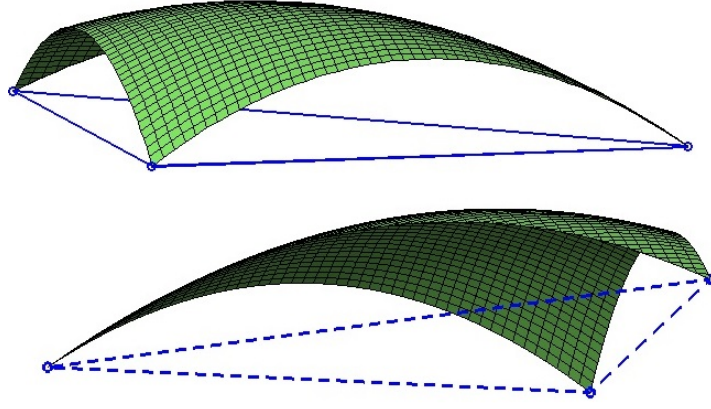


Figura 4.1: Parches triangulares cuadráticos.

Siguiendo esta notación, $\mathbf{b}(x) = \sum_{|\mathbf{i}|} \mathbf{b}_{\mathbf{i}} B_{\mathbf{i}}^2(\chi)$ es un parche triangular cuadrático definido por los puntos de control $\mathbf{b}_{\mathbf{i}}$. Similarmente a la situación de curvas de Bézier [Anexo C], las cuales interpolan a su primer y último punto de control, y al caso de superficies de Bézier [Anexo C], que interpolan los cuatro puntos frontera de su dominio, un parche triangular de Bézier interpola los vértices del triángulo de referencia, como se puede ver en la Figura 4.1. Los puntos de control diferentes a los vértices del triángulo de referencia se denominan puntos de control intermedios.

Análogamente a como se definen curvas de Bézier racionales, se definen también las superficies de Bézier racionales, y en particular, los parches de Bézier racionales. Dados puntos de control $\mathbf{b}_{\mathbf{i}}$ y pesos $w_{\mathbf{i}} \in \mathbb{R}$, correspondientes a cada punto de control, el parche triangular racional definido por esos pesos y puntos de control está dado por

$$\mathbf{b}(x) = \frac{\sum_{\mathbf{i}} w_{\mathbf{i}} \mathbf{b}_{\mathbf{i}} B_{\mathbf{i}}^n(\chi)}{\sum_{\mathbf{i}} w_{\mathbf{i}} B_{\mathbf{i}}^n(\chi)}$$

Así, los parches de la Figura 4.2 son parches triangulares cuadráticos racionales. Albrecht, Paluszny y Lentini en [1] dan las condiciones para que este tipo de parches yacza sobre una superficie cuádrica, es decir, sean parches cuádricos.

4.3. Un algoritmo intuitivo para la construcción de parches cuadráticos racionales que yacen sobre cuádricas

El número de puntos de control de una curva de Bézier racional determina su grado, y los pesos asociados a sus puntos de control alejan o acercan la curva hacia estos. Esto no solo determina la forma de la curva sino que, dependiendo de los valores de los pesos y la ubicación de los puntos de control, determina el cumplimiento de ciertas restricciones como tener a una línea prescrita como

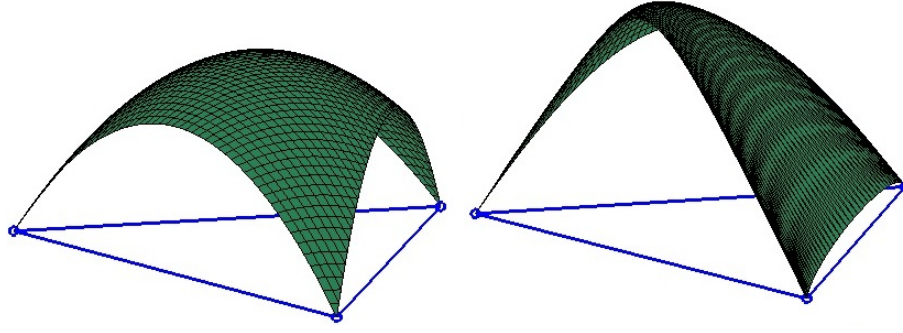


Figura 4.2: El parche izquierdo tiene todos sus pesos iguales a uno, mientras que en el parche de la derecha se modificaron los pesos correspondientes a algunos puntos de control.

tangente en un punto o interceptarla en varios puntos.

En la Figura 4.3, en azul, se observa la curva de Bézier racional definida por los puntos $\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3$, con pesos asociados iguales a uno. Si se quiere una curva cuadrática racional pero que además sea tangente a una recta prescrita (de color negro en la Figura 4.3) en un punto, se debe modificar el valor del peso asociado al punto de control \mathbf{b}_2 . La nueva curva definida por los puntos de control y los nuevos pesos, color magenta, cumple la condición de tangencia deseada.

En el caso de superficies de Bézier, en particular parches triangulares cuadráticos racionales, se puede pensar de forma análoga: al asignar ciertos pesos a los puntos de control, el parche definido por los puntos de control y estos pesos cumplirá algunas restricciones geométricas. Una de estas restricciones geométricas podría ser la de yacer sobre una cuádrica y cumplir condiciones de tangencia establecidas en los vértices dados.

Dado un triángulo $r_0r_1r_2$ y las normales en sus vértices, si la configuración formada por el triángulo y sus normales es una configuración de Brianchon, entonces el algoritmo presentado en [1] calcula los puntos de control intermedios, y sus respectivos pesos, los cuales junto a los vértices del triángulo definen un parche triangular cuadrático racional que yace sobre una cuádrica y cumple condiciones de tangencia en los vértices de $r_0r_1r_2$. Los pesos sobre cada vértice del triángulo son iguales a uno.

Cuando a partir del triángulo $r_0r_1r_2$ y las normales dadas no se obtiene una configuración de Brianchon, se realiza un proceso de optimización [1] para modificar las normales de tal manera que el nuevo sistema triángulo-normales permita la construcción de un parche que minimice la desviación de las condiciones de tangencia.

Algoritmo

Entrada: vértices (b_A, b_B, b_C) y normales en esos puntos (n_A, n_B, n_C) (Estos son datos 2D pues son las proyecciones de los vértices (b_1, b_2, b_3) y las normales (n_1, n_2, n_3) sobre el plano π) tal que la

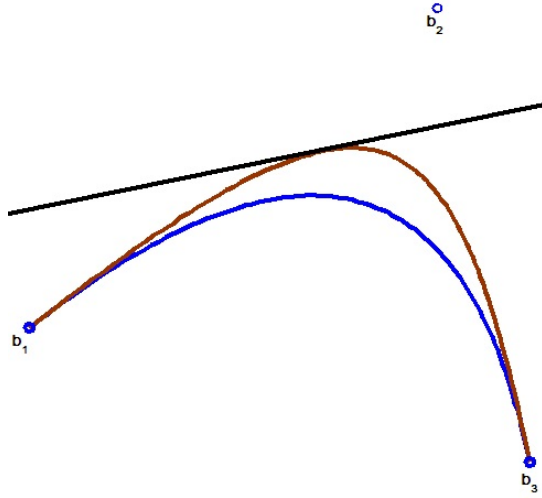


Figura 4.3: Curva de Bézier racional con diferentes pesos: para la azul los pesos son $(1, 1, 1)$ mientras para la curva magenta son $(1, 1.7, 1)$.

configuración vértices-normales sea una configuración de Brianchon.

Paso 1: Se construye el triángulo circunscrito $\tau(c_{AB}, c_{BC}, c_{CA})$ donde $b_A \in c_{CA}c_{AB}$, $b_B \in c_{AB}c_{BC}$, $b_C \in c_{BC}c_{CA}$.

Paso 2: Se calcula el punto p , exterior al plano π , intersección de los planos definidos por las normales en los vértices.

Paso 3: Se escogen los tres puntos de control intermedios $\mathbf{b}_{AB}, \mathbf{b}_{BC}, \mathbf{b}_{CA}$ sobre los lados del tetraedro formado por el triángulo \triangle_c y el punto p , como $\mathbf{b}_{AB} = 0.5(c_{AB} + p)$, $\mathbf{b}_{BC} = 0.5(c_{BC} + p)$ y $\mathbf{b}_{CA} = 0.5(c_{CA} + p)$

Paso 4: Se encuentra el punto de intersección s de los planos definidos por las ternas $(b_A, \mathbf{b}_{AB}, b_B)$, $(b_A, \mathbf{b}_{CA}, b_C)$ y $(b_B, \mathbf{b}_{BC}, b_C)$.

Paso 5: Se calculan los pesos w_{AB}, w_{BC}, w_{CA} asociados a los puntos de control intermedio como sigue:

$$w_{AB} = \frac{\lambda_{AB} \|b_B - b_A\|}{2\sqrt{\|d_{AB} - b_A\| \|b_B - d_{AB}\|}}$$

$$w_{BC} = \frac{\lambda_{BC} \|b_C - b_B\|}{2\sqrt{\|d_{BC} - b_B\| \|b_C - d_{BC}\|}}$$

$$w_{CA} = \frac{\lambda_{CA} \|b_A - b_C\|}{2\sqrt{\|d_{CA} - b_C\| \|b_A - d_{CA}\|}}$$

donde $\lambda_{AB}, \lambda_{BC}, \lambda_{CA}$ son escalares tales que

$$\lambda_{AB}(\mathbf{b}_{AB} - s) = d_{AB} - s$$

$$\lambda_{BC}(\mathbf{b}_{BC} - s) = d_{CB} - s$$

$$\lambda_{CA}(\mathbf{b}_{CA} - s) = d_{CA} - s$$

siendo d_{AB}, d_{BC} y d_{CA} los puntos de intersección del plano ϵ con las líneas $s\mathbf{b}_{AB}, s\mathbf{b}_{BA}$ y $s\mathbf{b}_{CA}$. Los pesos asociados a los vértices de entrada son iguales a uno.

Ejemplo 1

Se usa el algoritmo anterior para construir un parche a partir de los siguientes datos:

Vértices: $\{(1, 0.2718), (-0.7247, -0.7407), (-1, 0.2718)\}$ (proyecciones de $\{(2, 0, 0), (1, 1, \sqrt{2}), (1, 0, \sqrt{3})\}$)

Normales: $\{(1, 0.278), (-0.7247, -0.7407), (-1, 0.2718)\}$ (proyecciones de $\{(2, 0, 0), (1, 1, \sqrt{2}), (1, 0, \sqrt{3})\}$)

Estos vértices están sobre la esfera $x^2 + y^2 + z^2 = 1$, y las normales de entrada son las normales a la esfera en estos puntos, lo cual garantiza que la configuración dada por vértices y normales sea de Brianchon.

En la Figura 4.4 (izquierda) se observa, en azul, la proyección del triángulo t_N , definido por los vértices, en el plano π y, en negro, el triángulo τ definido por la intersección de los planos tangentes, definidos por las normales dadas, y el plano π .

Las líneas negras punteadas son las cevianas de la configuración, las cuales se intersecan en un punto, por tanto, la configuración mostrada es una configuración de Brianchon. Así, no es necesario ajustar las normales.

El parche triangular construido por el algoritmo a partir de la configuración se muestra en la Figura 4.4 (derecha).

No es muy usual encontrar configuraciones de Briachon, por lo que en la mayoría de los casos se debe aplicar el proceso de optimización para ajustar las normales dadas.

Ejemplo 2

Vértices: $\{(0.5774, 0.7071), (-1.1547, -0.7071), (1.1547, 0.7071)\}$ (proyecciones de $\{(1, 0, 0), (-1, 0, 1), (0, 1, 1)\}$)

Normales: $\{(0.5774, 0.7071), (0, 1.4142), (-0.5774, 0.7071)\}$ (proyecciones de $\{(1, 0, 0), (1, 1, 0), (0, 1, 0)\}$)

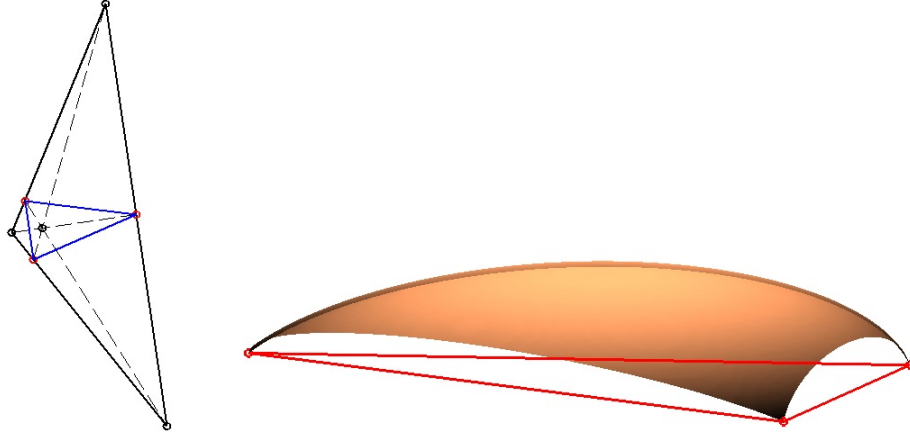


Figura 4.4: (Izquierda) Configuración de Brianchon. (Derecha) Parche triangular cuadrático racional que yace sobre una cuádrica, interpola a los vértices de entrada y tiene como planos tangentes en estos puntos los definidos por las normales de entrada

En la Figura 4.5 (izquierda) se observa la configuración triangular asociada a los datos de entrada. De nuevo, t_N es el triángulo azul, τ el triángulo negro y las líneas punteadas las cevianas. En este caso, las cevianas no se intersecan en un punto, por lo que la configuración triangular mostrada no es de Brianchon, por lo tanto es necesario ajustar las normales. El triángulo magenta es el triángulo que resulta de la intersección del plano π con los planos definidos por las nuevas tangentes. En la Figura 4.5 (derecha) se observa el parche construido a partir de la nueva configuración (triángulo magenta-triángulo azul).

Dada una malla triangular, se quiere obtener una malla reducida tal que sus triángulos, y respectivas normales, formen una configuración de Brianchon. Así, el algoritmo de reducción de mallas triangulares debe tener un procedimiento eficaz para decidir si un triángulo satisface o no la condición de Brianchon, ya que de satisfacerla, tal triángulo no debe ser eliminado en el proceso de reducción. Una posibilidad para hacer esto es usar el programa descrito anteriormente, sin embargo, esto podría resultar computacionalmente costoso por lo que diseñamos una heurística con la finalidad de clasificar los triángulos de la malla reducida. Para ello usamos el número de Ceva.

4.3.1. El número de Ceva: factor heurístico

A partir del Teorema 1.2 se definió el número de Ceva como

$$\zeta = \frac{AB}{BC} \cdot \frac{CD}{DE} \cdot \frac{EF}{FA}$$

para un triángulo ACE con cevianas BE, CF y DA como se muestra en la Figura 4.6.

Durante el desarrollo de la Tesis se hicieron experimentos con diferentes triangulaciones para medir

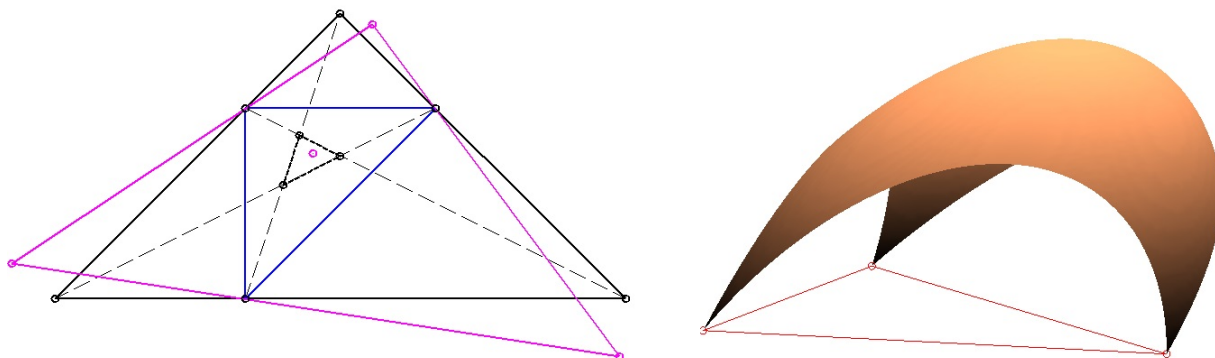


Figura 4.5: (Izquierda) Configuración triangular cercana a una configuración de Brianchon. (Derecha) Parche triangular a partir de la configuración dada.

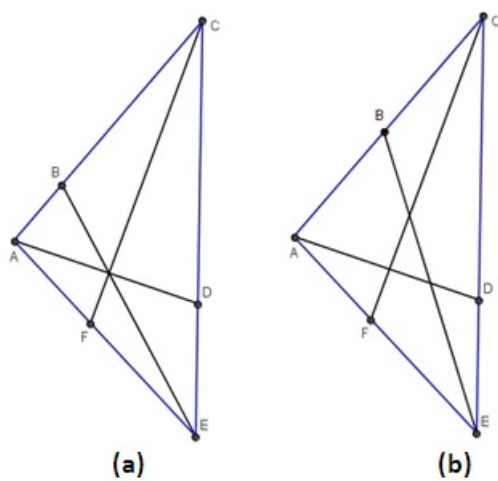


Figura 4.6: (a) Cevianas concurrentes. (b) Cevianas no concurrentes.

la capacidad del número de Ceva para determinar si un triángulo junto a sus normales permite la construcción de un parche triangular como los descritos en este capítulo.

En el algoritmo que modifica las normales para que la configuración vértices-normales sea una configuración de Brianchon aparece el número \mathbf{f} que mide la distancia entre la configuración inicial y la modificada. Si $\sqrt{\mathbf{f}} < 0.01$ se construyen parches que aproximan bastante bien las condiciones de tangencia impuestas por las normales dadas.

Los resultados de los experimentos realizados en MATLAB con programas creados durante el desarrollo de la Tesis se encuentran en la Tabla 4.1. A partir de estos experimentos, se escogió el valor de 1.5 como cota superior para el número de Ceva. Así, en más del 98 % de los casos, las configuraciones triángulos-normales que satisfacen $\zeta < 1.5$, satisfacen también $\sqrt{\mathbf{f}} < 0.01$, en cuyo caso se pueden construir los parches triangulares. Si se escoge 3 como cota superior, es posible construir parches triangulares en el 80 % de los casos.

En algunas mallas, el comportamiento del número de Ceva puede ser extraño, lo cual se debe a triángulos muy pequeños o cercanos a ser degenerados, por eso se recomienda usarlo como factor heurístico para mallas no muy grandes (< 200000 triángulos) o mallas reducidas a partir de una malla inicial.

Triangulación	$\zeta < 1.5$	$\zeta < 3$	$\sqrt{f} < 0.01$
Bunny1 (Muestra) Puntos: 35947 Triángulos: 69451 Triángulos muestra: 11859	3728 3728 (100 %)	7461 6121 (82 %)	8149
Bunny2 Puntos: 8171 Triángulos: 16301	3580 3553 (99, 2 %)	8235 7975 (96, 8 %)	13804
Bunny3 Puntos: 1889 Triángulos: 3851	915 909 (99, 3 %)	1995 1945 (97, 5 %)	3234
Bunny4 Puntos: 453 Triángulos: 948	240 240 (100 %)	504 495 (98, 2 %)	784
Bunny5 Puntos: 502 Triángulos: 1000	289 288 (99, 7 %)	571 556 (97, 4 %)	846
Dragon1 (Muestra) Puntos: 100250 Triángulos: 202520 Triángulos muestra: 33794	10452 10448 (99, 9 %)	21443 19618 (91, 5 %)	27092
Dragon3 Puntos: 5205 Triángulos: 11102	2233 2210 (98, 9 %)	5225 5053 (96, 7 %)	9036
Dragon4 Puntos: 506 Triángulos: 1000	193 192 (99, 5 %)	424 407 (96 %)	778
TEsf Puntos: 422 Triángulos: 840	660 660 (100 %)	771 771 (100 %)	840
TPeon Puntos: 429 Triángulos: 304	88 88 (100 %)	208 206 (99 %)	236

Tabla 4.1: Los datos en azul representan configuraciones con $\sqrt{f} < 0.01$. Así, para la triangulación **Bunny2**, de 3580 configuraciones con $\zeta < 1.5$, 3553 tienen $\sqrt{f} < 0.01$.

Bosquejo de un algoritmo de reducción y ajuste de mallas triangulares

Teniendo en cuenta lo planteado en este documento, se puede construir, a futuro en una investigación más avanzada, un algoritmo de reducción de mallas triangulares que genere mallas cuyos triángulos permitan la construcción de parches triangulares cuadráticos racionales que yacen sobre cuádricas, interpolan los vértices del triángulo y tienen como planos tangentes en estos puntos los planos definidos por las normales a la superficie en los vértices. Este algoritmo deberá estar enmarcado en el contexto general definido por Kobbelt et al en [5], estudiado en el Capítulo 2. Es decir, debe tener claramente definido un operador topológico, una noción de distancia y un criterio de calidad.

Este futuro algoritmo deberá tener los siguientes pasos

Primera parte

Entrada: Malla triangular en representación FV (Face-Vertex)

Paso 1. Lectura de datos.

Paso 2. Organización y clasificación de la malla.

Paso 3. Generación de la lista de prioridad.

Salida: Malla triangular en representación VV (Vertex-Vertex) y lista de prioridad.

Segunda parte

Entrada: Malla triangular en representación VV (Vertex-Vertex) y lista de prioridad

Paso 4. Reducción y ajuste de la malla triangular.

Los siguientes pasos se realizan sobre todos los vértices de la lista de prioridad:

Paso i. Aplicación del operador topológico. (reducción local)

Paso ii. Actualización de la información de la zona afectada por la reducción.

Paso iii. Actualización de la lista de prioridad.

Paso 5. Cálculo de la distancia aproximada entre las dos mallas triangulares.

Paso 6. Almacenamiento de la nueva malla triangular.

Salida: Malla triangular reducida en representación FV (Face-Vertex).

Tercera parte

Entrada: Malla triangular en representación FV (Face-Vertex).

Paso 7. Construcción de parches triangulares.

%%Comentario: Hasta este paso del algoritmo se puede llegar contando con la información plasmada en esta Tesis. Los siguientes pasos requieren una investigación de nivel más avanzado.

Paso 8. Pegado de parches.

Paso 9. Despliegue gráfico de la malla triangular reducida, cubierta por sus parches triangulares.

Salida: Parches triangulares que definen la malla reducida.

Se recomienda como operador topológico el colapso de medio lado, por su facilidad, y como criterio de calidad a $R(v)$ (redondez promedio), es decir, se recomienda se eliminen los vértices que tengan vecindades con triángulos no redondos. Así, la malla resultante tendrá triángulos que en un alto porcentaje cumplen las condiciones para la construcción de los parches requeridos. También se podría optar por incluir directamente al número de Ceva en el criterio de calidad.

Anexo A

Algoritmos tipo “greedy”

Un algoritmo tipo “greedy”(voraz) es aquel que, para resolver un determinado problema, sigue una heurística consistente en elegir la opción óptima local en cada paso con la esperanza de llegar a una solución óptima global. En muchos problemas, una estrategia de este tipo generalmente no produce una solución óptima, sin embargo, una heurística tipo greedy puede llevar a soluciones óptimas locales que aproximen una solución óptima global en un tiempo razonable.

Esquema

Dado un número finito de elementos C , un algoritmo greedy devuelve un conjunto $P \subset C$ de elementos seleccionados que cumple las restricciones del problema inicial. Cada conjunto P que satisfaga las restricciones se le denomina conjunto solución, y si además logra que la función objetivo se maximice o se minimice, según el caso, se denominará solución óptima. Un algoritmo tipo greedy tiene cinco componentes:

1. *Conjunto de candidatos* C .
2. *Función selección*: escoge el mejor candidato a agregar al conjunto solución, el cual no puede haber sido escogido con anterioridad.
3. *Función de factibilidad*: determina si a partir de un conjunto se puede llegar a una solución. Esta función se aplica al conjunto de seleccionados unido con el mejor candidato.
4. *Función solución*: comprueba si el conjunto de elementos seleccionados es una solución, sea óptima o no.
5. *Función objetivo*: es aquella que se quiere maximizar o minimizar.

En cada iteración se escoge el mejor candidato $x \in C$, se comprueba si $P \cup \{x\}$ es un conjunto factible, es decir, que al agregar x a P no se viole ninguna restricción establecida. En caso de producirse un conjunto factible, se agrega x a P . Por último, se comprueba si el conjunto de elementos seleccionados forma una solución. El algoritmo finaliza cuando se encuentre una solución o cuando se agoten los elementos de C .

Ejemplo

Dado el conjunto de nodos y aristas con peso mostrados en la Figura Ax1, se quiere escoger el camino más largo, de acuerdo al peso de cada arista, que inicie en A y pase por al menos otros tres nodos.

Análisis del problema

El conjunto de candidatos está dado por $C = \{A, B, \dots, O\}$. La función selección escogerá el mejor candidato basado en la conectividad entre el último nodo incluido al conjunto P y los nodos en $C \setminus P$, teniendo en cuenta también el peso de cada arista. La función de factibilidad comprobará si ciertas selecciones permiten llegar a una solución, así, si $P = \{A, B\}$, $P \cup F$ no es un conjunto factible pues no habría forma de agregar un cuarto nodo al conjunto ya que F no tiene descendientes. La función solución indicará si P cumple con las restricciones, en este caso, que contenga a A y a otros tres elementos, mientras la función objetivo contará la distancia recorrida, sumando el peso de las aristas que conforman el camino solución.

- Iteración 1: $P = \{\}$. Se escoge el nodo A como mejor candidato puesto que el camino debe iniciar con este. Se aplica la función de factibilidad al conjunto $P \cup \{A\}$, la cual determina que se podrá formar una solución a partir de este, ya que la altura del nodo A es cuatro. Se actualiza P a $P = \{A\}$ y se aplica la función solución, la cual establece que el conjunto P no es un conjunto solución.
- Iteración 2: $P = \{A\}$. Se escoge el nodo B como mejor candidato pues el peso de la arista que lo une con A es el óptimo local. Se aplica la función de factibilidad al conjunto $P \cup \{B\}$, la cual determina que se puede formar una solución a partir de éste conjunto. Se actualiza P como $P = \{A, B\}$ y se aplica la función solución, la cual establece que aún no se ha formado un conjunto solución.
- Iteración 3: $P = \{A, B\}$. Se escoge como mejor candidato al nodo F , pero a partir de $P \cup \{F\}$ no se puede formar una solución, así que se escoge como mejor candidato al nodo D . $P \cup \{D\}$ resulta ser un conjunto factible, por lo tanto se actualiza el conjunto de elementos seleccionados como $P = \{A, B, D\}$. P aún no es un conjunto solución.

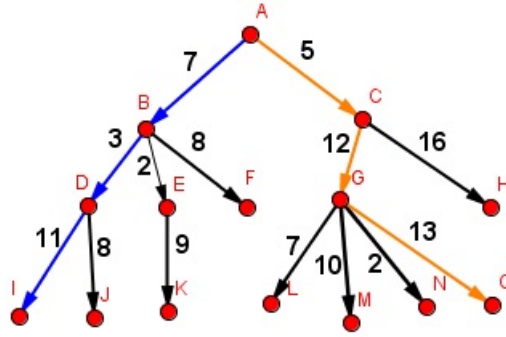


Figura A.1: Camino greedy (azul) vs Camino más óptimo (naranja).

- Iteración 4: $P = \{A, B, D\}$. Se escoge como mejor candidato a I . $P \cup \{I\}$ es un conjunto factible y $P = \{A, B, D, I\}$ una solución para el problema. La función objetivo calcula que el camino más largo escogido con el anterior procedimiento tipo greedy es 21.

Como ya se mencionó, un algoritmo tipo greedy no siempre da la solución óptima global del problema, esto se puede observar en la Figura A.1, pero lleva a una solución aproximada en un tiempo razonable.

Anexo B

Geometría diferencial de superficies

La geometría diferencial de superficies es una rama de la geometría diferencial que propone métodos y definiciones para estudiar superficies que poseen cierto tipo de estructuras, siendo la más común, una métrica riemanniana. Las superficies han sido ampliamente estudiadas desde diferentes perspectivas: *extrínsecamente*, relativo a su embebimiento en el espacio euclideo, e *intrínsecamente*, reflejando sus propiedades determinadas únicamente por las distancias dentro de la superficie, medidas a lo largo de curvas de la misma. Uno de los principales conceptos es el de curvatura. Gauss demostró que la curvatura es una propiedad intrínseca de la superficie que no depende del empotramiento de ésta en el espacio euclideo.

Ecuación paramétrica de una superficie

Sea S una superficie en \mathbb{R}^3 , una representación paramétrica para S está dada por

$$r(u, v) = (x(u, v), y(u, v), z(u, v))$$

donde x , y y z son funciones de dos variables y u y v están en un dominio D . Las primeras derivadas parciales de esta función se representan como $r_u = \frac{\partial r}{\partial u}$ y $r_v = \frac{\partial r}{\partial v}$.

Plano tangente a la superficie en un punto

Dada una superficie S de \mathbb{R}^3 y un punto $p_0 = (u_0, v_0) \in S$. p_0 es un punto regular si $r_u(u_0, v_0) \times r_v(u_0, v_0) \neq 0$.

El plano tangente a la superficie en un punto regular $p_0 = (u_0, v_0)$ es el plano afín en \mathbb{R}^3 expandido por los vectores $r_u(u_0, v_0)$, $r_v(u_0, v_0)$ y que pasa por p_0 . Así, un vector normal unitario para este plano tangente es

$$n = \frac{r_u(u_0, v_0) \times r_v(u_0, v_0)}{\|r_u(u_0, v_0) \times r_v(u_0, v_0)\|}$$

El plano tangente a la superficie en p_0 no interseca a la superficie (localmente) en otro punto si la superficie es de curvatura gaussiana positiva. Si la superficie es de curvatura gaussiana negativa o cero entonces el plano interseca a la superficie en otro punto.

Primera forma fundamental

La primera forma fundamental es el producto interno sobre el espacio tangente de una superficie en un espacio tridimensional euclideo y se denota por $I(v, w) = \langle v, w \rangle$. Gracias a la primera forma fundamental se puede estimar longitudes de curvas sobre la superficie, ángulos de intersección entre curvas entre otros conceptos métricos.

Sea $r(u, v)$ una superficie paramétrica. El producto interno de dos vectores del espacio tangente está dado por

$$I(ar_u + br_v, cr_u + dr_v) = ac\langle r_u, r_u \rangle + (ad + bc)\langle r_u, r_v \rangle + bd\langle r_v, r_v \rangle = Eac + F(ad + bc) + Gbd$$

donde $E = r_u \cdot r_u$, $F = r_u \cdot r_v$ y $G = r_v \cdot r_v$.

Longitud de una curva

Sea C una curva contenida en la superficie S , con ecuación paramétrica

$q(t) = (x(u(t), v(t)), y(u(t), v(t)), z(u(t), v(t)))$. La longitud de la curva C está dada por

$$L_C = \int_C \sqrt{Eu'(t)^2 + 2Fu'(t)v'(t) + Gv'(t)^2} dt$$

Ángulo entre curvas

Dadas dos curvas paramétricas C_1 y C_2 que se intersecan en un punto p_0 y cuyas ecuaciones están dadas por

$$\begin{aligned} q_1(t) &= (x(u_1(t), v_1(t)), y(u_1(t), v_1(t)), z(u_1(t), v_1(t))) \\ q_2(t) &= (x(u_2(t), v_2(t)), y(u_2(t), v_2(t)), z(u_2(t), v_2(t))) \end{aligned}$$

el ángulo δ formado por las dos curvas en el punto de intersección se define como

$$\cos \delta = \frac{Eu'_1u'_2 + F(u'_1v'_2 + u'_2v'_1) + Gv'_1v'_2}{\sqrt{E(u'_1)^2 + 2F(u'_1v'_1 + G(v'_1)^2)} \sqrt{E(u'_2)^2 + 2F(u'_2v'_2 + G(v'_2)^2)}}$$

donde las derivadas se evalúan en parámetros t_1 y t_2 tales que $p_0 = q_1(t_1) = q_2(t_2)$.

Segunda forma fundamental

La segunda forma fundamental es una forma cuadrática sobre el plano tangente de una superficie suave en un espacio euclideo tridimensional, usualmente denotado por II . Junto con la primera forma fundamental determina la curvatura de curvas sobre la superficie. Bajo el mismo contexto, siendo S una superficie y $r = r(u, v)$ su parametrización, la segunda forma fundamental está definida por

$$II = Ldu^2 + 2Mdudv + Ndv^2$$

donde $L = r_{uu} \cdot n$, $M = r_{uv} \cdot n$ y $N = r_{vv} \cdot n$.

Curvaturas

Una curva sobre una superficie puede ser vista como una curva en \mathbb{R}^3 , donde le son aplicables tanto las fórmulas de la geometría diferencial de curvas como las de la geometría diferencial de superficies. Lo anterior permite relacionar la curvatura total de la curva con la curvatura medida desde la superficie. La curvatura total X_q de una curva $q(t)$ puede ser descompuesta entre una componente tangencial a la superficie (medible dentro de la misma) llamada **curvatura geodésica** k_g y una componente perpendicular a la superficie (que depende de cómo está curvada la superficie en el espacio y cuál sea la dirección de la curva dentro de la superficie) llamada **curvatura normal** k_n . Estas curvaturas satisfacen $X_q^2 = k_g^2 + k_n^2$.

Las curvaturas normal y geodésica de una curva sobre una superficie pueden calcularse fácilmente a partir del vector tangente a la curva y las normales a la curva y a la superficie:

$$k_n = \frac{II(q'(t), q'(t))}{I(q'(t), q'(t))}, \quad k_g = \frac{n \cdot (q'(t) \times q''(t))}{\|q'(t)\|^3}$$

Las curvaturas principales se definen como las raíces k_1, k_2 de la ecuación cuadrática

$$\det(II - kI) = \det \begin{pmatrix} L - kE & M - kF \\ M - kF & N - kG \end{pmatrix}$$

donde la **curvatura gaussiana** $K = k_1 k_2$ y la **curvatura media** $H = (k_1 + k_2)/2$ se calculan como

$$K = \frac{LN - M^2}{EG - F^2}, \quad H = \frac{EN - 2FM + GL}{2(EG - F^2)}$$

Estas cantidades son independientes de la parametrización que se use, formando así importantes herramientas para el estudio de la geometría de la superficie. La curvatura media y las curvaturas principales cambian de signo de acuerdo a la orientación de la superficie, mientras la curvatura

gaussiana es totalmente independiente de la parametrización.

El signo de la curvatura gaussiana en un punto determina la forma de la superficie cerca al punto: si $K > 0$ la superficie es localmente convexa y el punto es llamado elíptico, si $K < 0$ la superficie es parecida a una silla de montar cerca al punto y éste es llamado hiperbólico, mientras si $K = 0$, el punto es llamado parabólico.

Anexo C

Curvas y superficies de Bézier

Curvas de Bézier

Dados $n + 1$ puntos $\mathbf{b}_0, \mathbf{b}_1, \dots, \mathbf{b}_n$ en el espacio, llamados puntos de control, y los polinomios de Bernstein $B_i^n(u)$ dados por

$$B_i^n(u) = \frac{n!}{i!(n-i)!} u^i (1-u)^{n-i}, \quad u \in [0, 1], \quad 0 \leq i \leq n$$

la curva de Bézier definida por los puntos de control $\mathbf{b}_0, \mathbf{b}_1, \dots, \mathbf{b}_n$ viene dada por

$$c(u) = \sum_{i=0}^n B_i^n(u) \mathbf{b}_i$$

La línea poligonal formada por la unión de los lados $\mathbf{b}_0\mathbf{b}_1, \mathbf{b}_1\mathbf{b}_2, \dots, \mathbf{b}_{n-1}\mathbf{b}_n$ es usualmente llamada línea poligonal de control o polígono de control.

Las curvas de Bézier tienen, entre otras, las siguientes propiedades:

- El grado de una curva de Bézier definida por $n + 1$ puntos de control es n .
- La curva pasa por los puntos de control \mathbf{b}_0 y \mathbf{b}_n .
- La suma de los polinomios de Bernstein, evaluados en un punto fijo u , es 1.
- La curva de Bézier yace completamente en la cápsula convexa definida por sus puntos de control.

Si el dominio de u es $[a, b]$ en vez de $[0, 1]$ se debe hacer el siguiente cambio de variable $\bar{u} = \frac{u-a}{b-a}$, de donde los polinomios de Bernstein estarían dados por

$$B_i^n(\bar{u}) = B_i^n\left(\frac{u-a}{b-a}\right) = \frac{n!}{i!(n-i)!} \left(\frac{u-a}{b-a}\right)^i \left(1 - \frac{u-a}{b-a}\right)^{n-i}$$

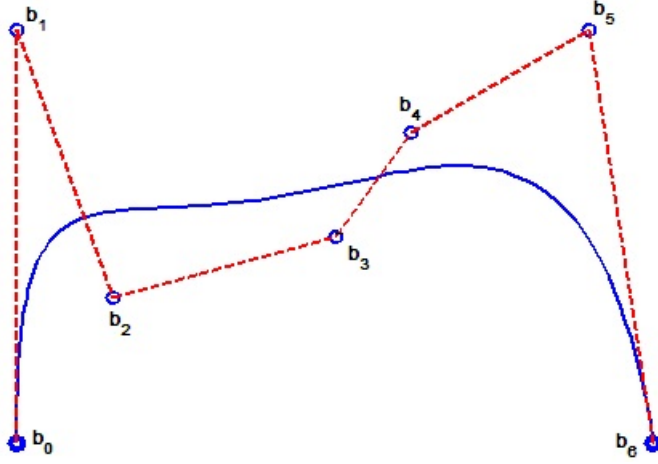


Figura C.1: Curva de Bézier, en azul, y su respectivo polígono de control, en rojo.

En la definición del polinomio de Bernstein se usa la convención $0^0 = 1$.

Curvas de Bézier racionales

Representaciones paramétricas que usan polinomios no son suficientemente poderosas ya que muchas curvas, por ejemplo, círculos, elipses e hipérbolas no pueden ser obtenidas de esta manera. Debido a esto se introducen las curvas racionales, las cuales resultan ser adecuadas para la representación de secciones cónicas.

Una curva de Bézier racional definida por los $n+1$ puntos de control $\mathbf{b}_0, \mathbf{b}_1, \dots, \mathbf{b}_n$ y los pesos no negativos w_0, w_1, \dots, w_n , asociados a cada punto de control, está dada por

$$c(u) = \sum_{i=0}^n R_i^n(u) \mathbf{b}_i$$

donde $R_i^n(u)$ se define como

$$R_i^n(u) = \frac{w_i B_i^n(u)}{\sum_{i=0}^n w_i B_i^n(u)}, \quad u \in [0, 1], \quad 0 \leq i \leq n$$

Las curvas de Bézier racionales poseen las mismas propiedades de las curvas de Bézier descritas anteriormente. Los pesos w_i actúan como parámetros de forma ya que su modificación aleja o acerca la curva hacia el punto de control correspondiente.

Superficies de Bézier

Una superficie de Bézier está definida por un conjunto de puntos de control \mathbf{b}_{ij} , de dimensión dos, donde $0 \leq i \leq m, 0 \leq j \leq n$:

$$s(u, v) = \sum_{i=0}^m \sum_{j=0}^n B_i^m(u) B_j^n(v) \mathbf{b}_{ij}$$

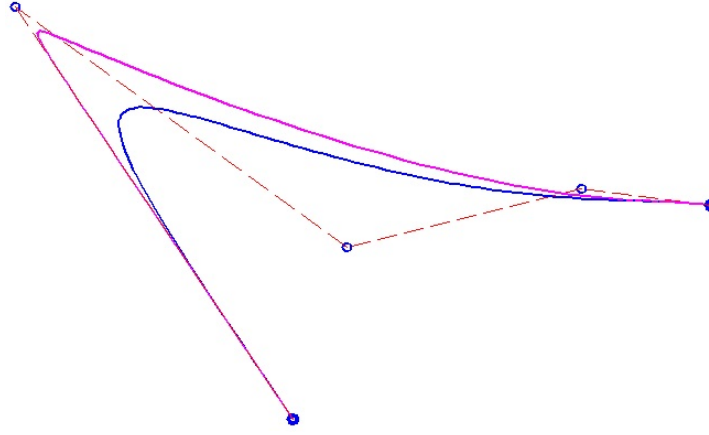


Figura C.2: La curva azul es la curva racional de Bézier definida por los puntos de control con pesos $w_i = 1$. La curva magenta es la curva racional de Bézier definida por los mismos puntos de control pero con uno de sus pesos igual a 10.

Como $B_i^m(u)$ y $B_j^n(v)$ son funciones de grado m y n respectivamente, la superficie $s(u, v)$ se llama superficie de Bézier de bigrado (m, n) . El conjunto de puntos de control se denomina malla de Bézier o malla de control. Si el dominio de las variables u y v es $[0, 1]$ entonces la superficie de Bézier mapea el cuadrado unitario en un parche rectangular.

Las superficies de Bézier son una generalización de las curvas de Bézier, y por tanto cumplen propiedades similares a las listadas para curvas:

- La superficie de Bézier pasa por los puntos de control de las cuatro esquinas de la malla de control: \mathbf{b}_{00} , \mathbf{b}_{m0} , \mathbf{b}_{0n} y \mathbf{b}_{mn}
- $\sum_{i=0}^m \sum_{j=0}^n B_i^m(u) B_j^n(v) = 1$, para todo u y v en $[0, 1]$
- La superficie yace en la cápsula convexa definida por la malla de control

Curvas isoparamétricas

Dado que una superficie paramétrica se define como $f(u, v) = (x(u, v), y(u, v), z(u, v))$, una curva isoparamétrica en la dirección de u para un valor fijo $v = t$ se denota como $f(u, t)$. De forma análoga se definen las curvas paramétricas en la dirección de v para un valor fijado de u . De forma más sencilla, las curvas isoparamétricas son llamadas u -curvas o v -curvas.

Sea $v = t$, entonces

$$s(u, t) = \sum_{i=0}^m \sum_{j=0}^n B_i^m(u) B_j^n(t) \mathbf{b}_{ij} = \sum_{i=0}^m B_i^m(u) \left(\sum_{j=0}^n B_j^n(t) \mathbf{b}_{ij} \right) = \sum_{i=0}^m B_i^m(u) \mathbf{b}_{ti}$$

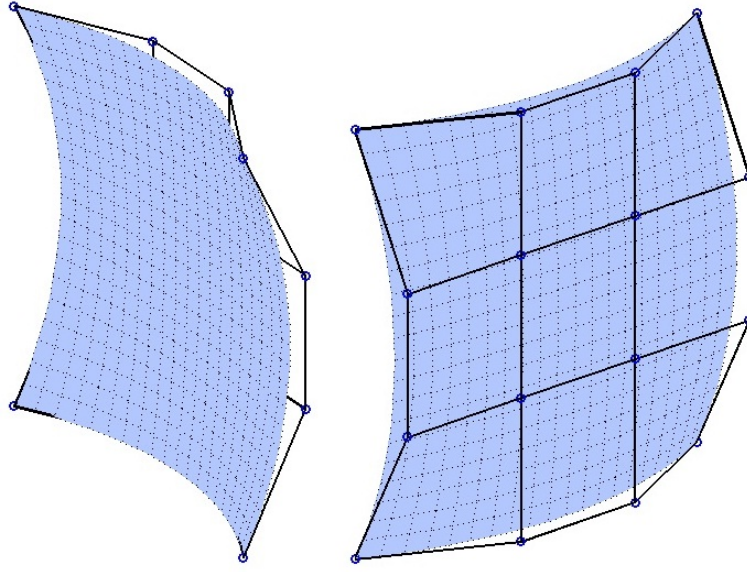


Figura C.3: Superficie de Bézier (azul) definida por su malla de control (negro). Sobre la superficie y con líneas puntadas están las curvas isoparamétricas.

donde $\mathbf{b}_{t_i} = \sum_{j=0}^n B_j^n(t) \mathbf{b}_{ij}$

Así, una u -curva de grado m es una curva de Bézier definida por $m + 1$ puntos de control \mathbf{b}_{t_i} que pueden hallarse a partir de la ecuación de la superficie. Las u -curvas y v -curvas cubren toda la superficie, como se puede observar en la Figura C.3.

Entre las curvas isoparamétricas destacan las curvas definidas por $s(u, 0)$, $s(u, 1)$, $s(0, v)$ y $s(1, v)$, las cuales son las curvas de Bézier que forman la frontera de la superficie de Bézier.

Bibliografía

- [1] Albrecht G., Paluszny M. y Lentini M., *An Intuitive Way for Constructing Parametric Quadric Triangles*. Enviado a publicación, 2012.
- [2] Ciampalini A., Cignoni P., Montani C. y Scopigno R., *Multiresolution Decimation Based on Global Error*, The Visual Computer, 1997, **13**, 228-246.
- [3] Huang C-M. y Tseng Y-H., *Plane Fitting Methods of LiDAR Points Cloud*, AARS, 2008
- [4] Klein R., Liebich G. y Straßer W., *Mesh Redution with Error Control*, IEEE Visualization '96, Conference Proceedings, 1996, 311-318.
- [5] Kobbelt L., Campagna S. y Seidel H.-P., *A General Framework for Mesh Decimation*. Computer Sciences Department, University of Erlangen-Nürnberg, 1998.
- [6] Paluszny M., Prautzsch H. y Boehm W., *Métodos de Bézier y B-Splines*, Universitätsverlag Karlsruhe, 2005.
- [7] Schroeder W.J., Zarge J.A. y Lorensen W.E., *Decimation of Triangle Meshes*. General Electric Company, 1992.
- [8] Schroeder W.J., *A Topology Modifying Progressive Decimation Algorithm*, GE Corporate RD Center, 1997.
- [9] Garland M. y Heckbert P.S., *Surface simplification using quadric error metric.*, SIGGRAPH'97, 209-216, 1997.
- [10] Hussain M., Okada Y. and Niijima K., *Efficient and Feature-Preserving Triangular Mesh Decimation*, Journal of WSCG, **12**, No.1-3., ISSN 1213-6972, WSCG'2004, 2004.
- [11] *Introduction to Computing with Geometry Notes* (20 de octubre de 2012). Shene C.-K. Department of Computer Science Michigan Technological University.
<http://www.cs.mtu.edu/shene/COURSES/cs3621/NOTES/>

- [12] *The Stanford 3D Scanning Repository* (5 de septiembre de 2012).
<http://graphics.stanford.edu/data/3Dscanrep/>
- [13] *Geometría diferencial de superficies* (14 de enero de 2013). Wikipedia La enciclopedia libre.
es.wikipedia.org/wiki/Geometría_diferencial_de_superficies